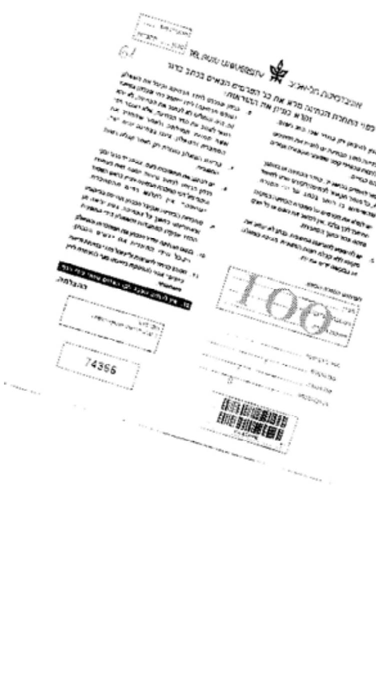
אגודת הסטודנטים
אוניברסיטת ת"א

ועד הנדסה

מבני נתונים

(לתלמידי תוכנה)

0368.2158



בהצלחה,

ועד הנדסה

שיטת האב – Master Method

$$T(n) = aT(\frac{n}{b}) + f(n)$$

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & if: \exists \varepsilon > 0, f(n) = O(n^{\log_b a - \varepsilon}) \\ \Theta(n^{\log_b a} \lg n) & if: f(n) = \Theta(n^{\log_b a}) \\ \Theta(f(n)) & if: \exists \varepsilon > 0, f(n) = \Omega(n^{\log_b a + \varepsilon}) \\ & and: \exists c < 1, a \cdot f(\frac{n}{b}) \leq c \cdot f(n) \end{cases}$$

מספר תעודת זהות: ____

מספר מחברת: ____

מספר קורס: 0368.2158

סמסטר א׳, מועד א׳, תשס״ח

01/05/ 2008

מבחן במבני נתונים

פרופ׳ אורי צוויק, פרופ׳ חיים קפלן, דן פלדמן, ליאור שפירא

משך המבחן שלוש וחצי שעות **(לא תינתן הארכה)**

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

הוראות כלליות:

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
4. **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.
5. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות. **תשובה ללא נמוק במקום שנדרש נמוק לא תזכה בנקודות.**
6. סיבוכיות זמני ריצה יש לכתוב כחסם הדוק ככל שתוכלו, בכתיב $O(\ )$.

בהצלחה!!

שאלה	ציון	ניקוד מקס׳
1		14
2		14
3		15
4		15
5		14
6		14
7		14
סה״כ		100

שאלה 1 (14 נקודות)

שאלה זו מתייחסת למבנה הנתונים union-find שהוצג בכתה. כל קבוצה מיוצגת על ידי עץ כאשר משתמשים ב-union by rank ו-path compression.

א. הוכח/י או הפרך/י: לאחר שבצענו פעולת find על איבר מסוים x, כל פעולת find נוספת על x תיקח זמן קבוע (במקרה הגרוע ביותר).

ב. כמה זמן נדרש, במקרה הגרוע ביותר, לביצוע סדרה בת m פעולות על n איברים שבה כל פעולות ה-union מבוצעות **לפני** כל פעולות ה-find? (הסבר/י את תשובתך.)

ג. תאר/י בקצרה כיצד ניתן להוסיף למבנה הנתונים פעולה rep(x) שהופכת את האיבר x לאיבר המייצג בקבוצה שבה הוא נמצא כרגע. מה סיבוכיות הפעולה במקרה הגרוע ביותר ובמקרה ה-amortized? (כאשר קבוצתו של x מתאחדת עם קבוצה אחרת, האיבר x יכול להפסיק להיות איבר מיצג.)

שאלה 2 (14 נקודות)

אנו מעוניינים בהכללה של מחסנית התומכת בנוסף לפעולות push ו-pop גם בפעולת minimum שמחזירה את האיבר הקטן ביותר במחסנית ובפעולת delete-min המסירה את האיבר הקטן ביותר מהמחסנית.

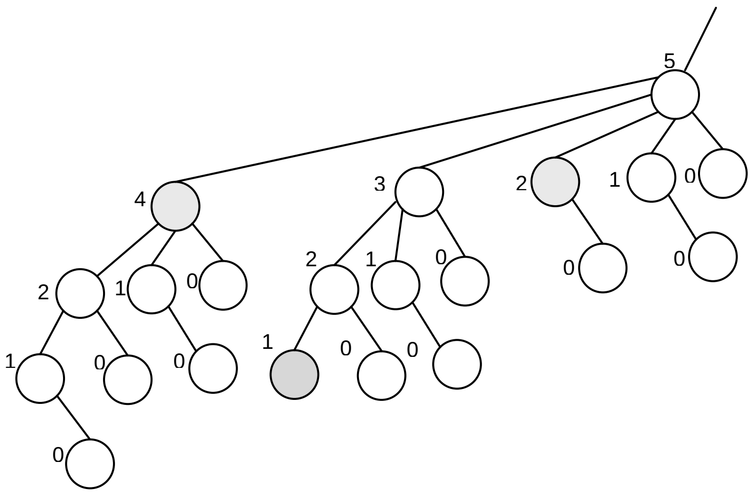
א. תאר/י מבנה נתונים התומך בפעולת minimum בזמן קבוע ובכל שאר הפעולות בזמן $O(\log n)$, כאשר n הוא מספר האיברים במחסנית בזמן ביצוע הפעולה. (כל הזמנים כאן הם במקרה הגרוע ביותר.) הוכח/י שהמימוש שתיארת הוא נכון ורץ בזמן הנדרש.

ב. תאר/י מבנה נתונים, שבמקרה ה-amortized, תומך בפעולות pop ו-delete-min בזמן $O(\log n)$ ובפעולות push ו-minimum בזמן $O(1)$. (כל הזמנים כאן הם amortized.) הוכח/י שהמימוש שתיארת הוא נכון ורץ בזמן הנדרש.

ג. הוכח/י שלא קיים במודל ההשוואות מבנה נתונים שתומך בכל ארבעת הפעולות push, pop, minimum, delete-min בזמן amortized קבוע.

<u>שאלה 3 (15 נקודות)</u>

נשנה את ערמת פיבונצי באופן הבא. העצים בערמה יהיו עצים בינומיים אולם לכל צומת שאינו שורש יתכן וחסר הבן השמאלי ביותר. כלומר לצומת מדרגה r יש r או r-1 בנים, מדרגות 0,1,...,r-1 או 0,1,...,r-2 בהתאמה. במידה ולצומת מדרגה r יש רק r-1 ילדים נקרא לו צומת **<u>חסר</u>**. שים לב כי ניתן לדעת אם צומת הוא חסר או לא על ידי השוואת דרגתו לדרגת בנו השמאלי ביותר. אם ההפרש הוא 1 , זהו צומת רגיל, אחרת זהו צומת חסר. הנה דוגמא לתת עץ של עץ בינומי חסר, ליד כל צומת רשומה דרגתו, הצמתים הצבועים באפור הם החסרים.



נשנה את הממוש של decrease-key (x,h,Δ) באופן הבא. אם ערכו החדש של x קטן מזה של אביו, ננתק את x מאביו ותת העץ ששורשו x יהפוך לעץ ביער. אם דרגתו של x היתה b אזי ל-w=p[x], אביו של x, חסר עתה בן מדרגה b. נתקן זאת על פי אחד מהמקרים הבאים (ראה ציור בעמוד הבא):

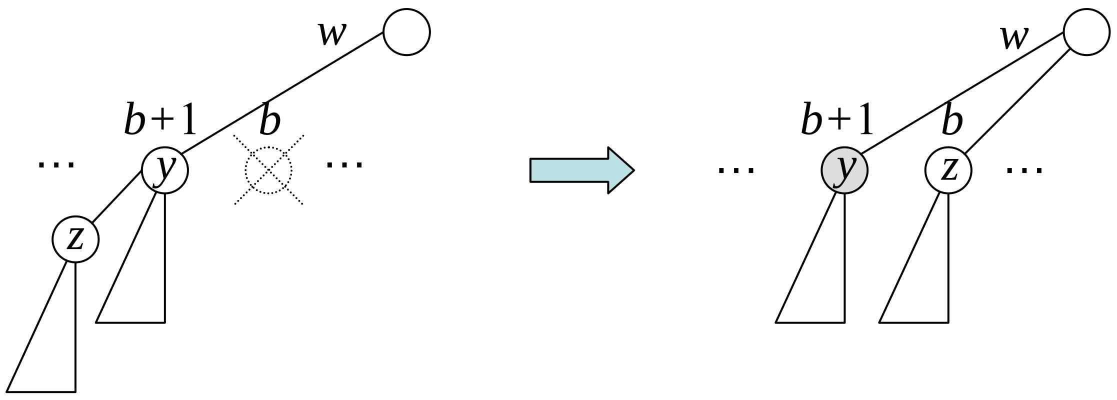
A. <u>ל-w יש בן y מדרגה b+1 ו-y אינו חסר:</u> יהא z הבן השמאלי ביותר של y. הצומת z הוא מדרגה b. ננתק אותו מ-y, שיהפוך להיות צומת חסר, ונהפוך אותו לבן מדרגה b של w. פעולת התיקון נגמרה.

B. <u>ל-w יש בן y מדרגה b+1 ו-y חסר:</u> נהפוך את y לצומת לא חסר מדרגה b. ל-w חסר עתה בן מדרגה b+1. נמשיך בתהליך התיקון עבורו.

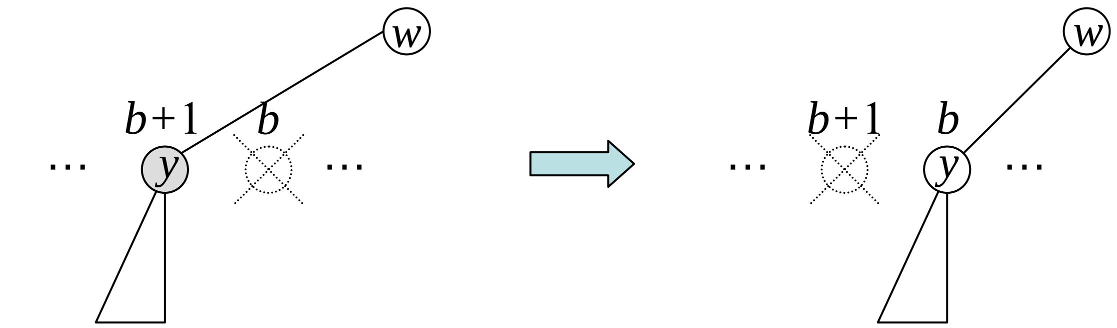
C. <u>ל-w אין בן y מדרגה b+1 ו-w אינו חסר:</u> w יהפוך לצומת חסר (מדרגה b+1) ופעולת התיקון הסתיימה.

D. <u>ל-w אין בן y מדרגה b+1 ו-w חסר:</u> w יהפוך לצומת לא חסר (מדרגה b). אם w איננו שורש, ננתק את w מאביו ונמשיך בפעולת התיקון עבור p[w] שחסר עתה צומת מדרגה b+2.

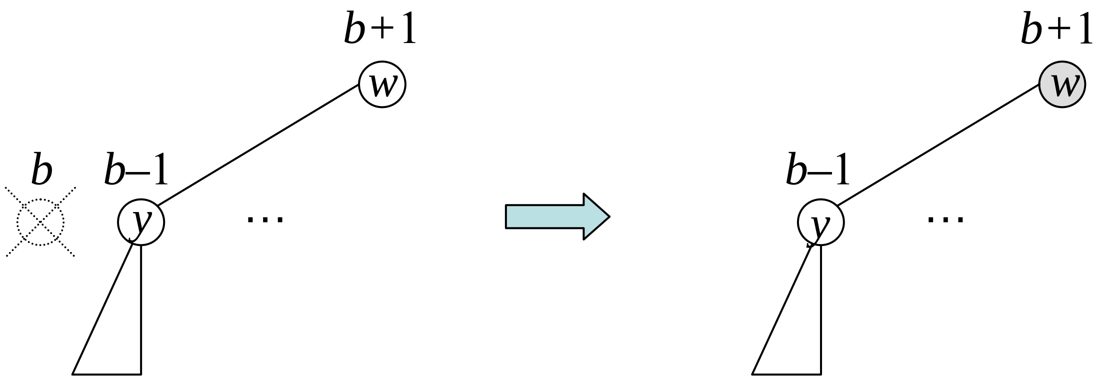
A



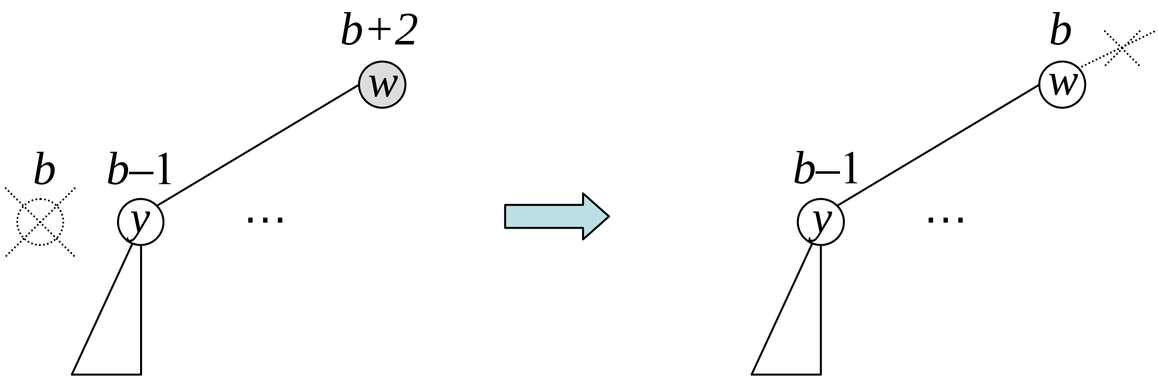
B



C



D



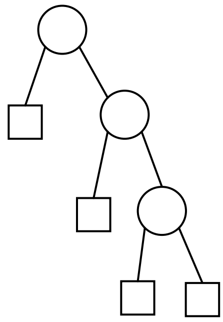
א. הוכח/י כי מספר הצמתים בעץ בינומי חסר מדרגה k הוא לפחות F_{k+1} כאשר $F_1=F_2=1$ ו-$F_k=F_{k-1}+F_{k-2}$ עבור $k \geq 3$. הסק/י שדרגתו של עץ בינומי חסר עם n צמתים היא $O(\log n)$.

ב. מהו הזמן במקרה הגרוע של פעולת ה-decrease-key ? הוכח/י את תשובתך.

ג. מהו הזמן amortized שלוקחת כל פעולה ? שים/י לב שמימוש שאר הפעולות זהה לחלוטין למימושן בערמות פיבונצי ? הוכח/י תשובתך. (רמז: חשוב על צומת חסר כצומת מסומן).

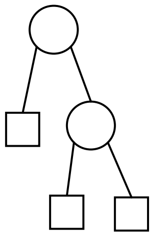
<u>שאלה 4 (15 נקודות)</u>

א. צבע/י כל אחד מהצמתים בעצים הבאים באדום או שחור כך שהעץ יהפוך לעץ שחור-אדום חוקי או הוכח/י שאין צביעה כזו. (סמן/י B עבור צומת שחור ו-R עבור צומת אדום.)



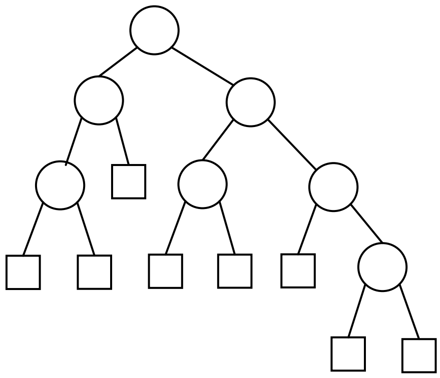
צביע / לא צביע

<u>הוכחה:</u>



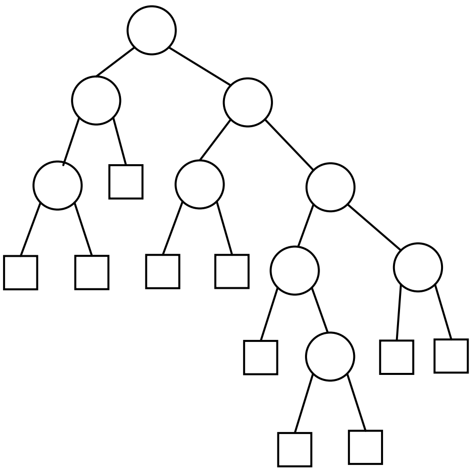
צביע / לא צביע

<u>הוכחה:</u>



צביע / לא צביע

<u>הוכחה:</u>



צביע / לא צביע

<u>הוכחה:</u>

במימוש עצי אדום-שחור (Red-black trees) אנו משתמשים, בין השאר, בפעולות מהסוגים הבאים:

R	קריאת מצביע	לדוגמא: left[x]
W	השמה למצביע	לדוגמא: left[x]←y
A	קריאת צבע של צומת	לדוגמא: color[x]
B	השמת צבע	לדוגמא: color[y] ←RED
C	השוואת שני מפתחות	לדוגמא: if k<key[x]

בניתוח שהוצג בכיתה הנחנו שעלות כל הפעולות האלו, וכן כל הפעולות הבסיסיות האחרות היא קבועה. נניח עתה שלכל אחת מחמשת הפעולות האלו עלות שונה. נניח שקריאת ערכו של מצביע דורשת R יחידות זמן, השמה למצביע דורשת W יחידות זמן, קריאת צבע דורשת A יחידות זמן, וכך הלאה. נניח גם שעלות כל פעולה אחרת היא 0.

לדוגמא, ביצוע סיבוב (rotation) דורש קריאה ושינוי של מספר קבוע של מצביעים. לכן עלות הפעולה היא $O(R+W)$. נשים לב ש-A,B ו-C אינם מופיעים בביטוי זה שכן ביצוע סיבוב אינו דורש קריאה וכתיבה של צבעים ואינו דורש השוואת מפתחות.

לנוחיותכם מצורפים בהמשך המקרים השונים בהם נתקלים בעת פעולת insert.

ב. עלות הפעולות find, insert במקרה הגרוע ביותר, במונחים של R,W,A,B,C וכן במונחי n, מספר האיברים בעץ בעת ביצוע הפעולה הוא:

	Worst-case
find	O( )
insert	O( )

בניתוח הפעולה insert הנח/י שמקום ההכנסה נתון ולכל צומת x מצביע p[x] להורהו.

הוכחה:

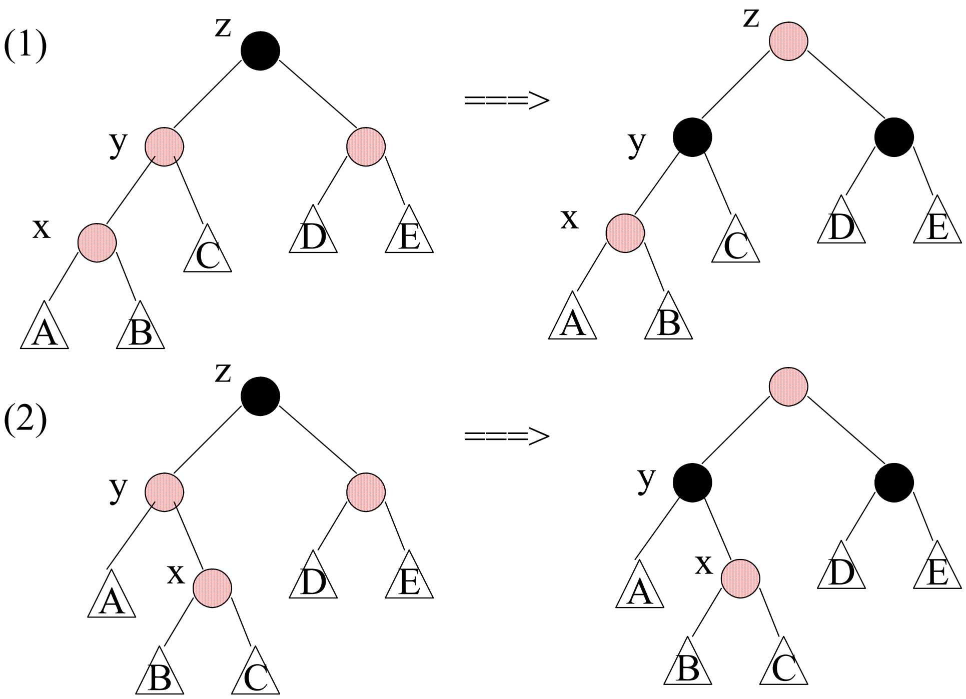
ג. עלות הפעולות find, insert במקרה ה-amortized, במונחים של R,W,A,B,C וכן במונחי n, מספר האיברים בעץ בעת ביצוע הפעולה הוא:

	Amortized
find	O( )
insert	O( )

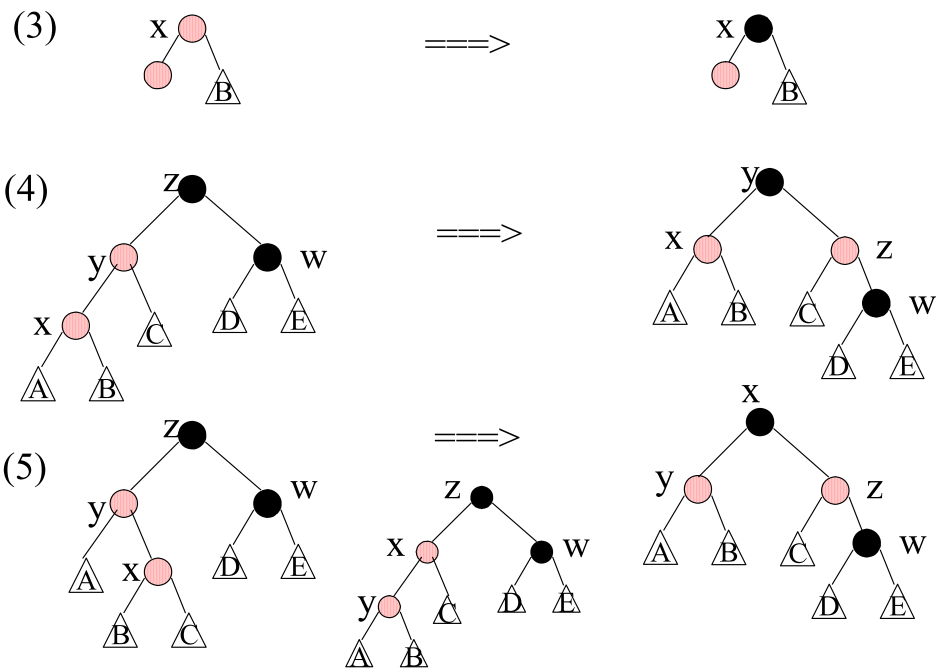
בניתוח הפעולות insert ו-delete הנח/י שמקום ההכנסה/הוצאה נתון ולכל צומת x מצביע p[x] להורהו.

הוכחה: (ציין/י במפורש את פונקצית הפוטנציאל בה את/ה משתמש.)

Insert -- non terminal cases



Insert -- terminal cases



<u>שאלה 5</u> (14 נקודות)

א. צייר/י Generalized Suffix Tree המייצג את המחרוזות 01100 ו-101101.

ב. תאר/י אלגוריתם יעיל ככל האפשר שמקבל מחרוזת בינארית s באורך n ומוצא את המחרוזת הארוכה ביותר p כך שגם p וגם המשלים של p הן תתי מחרוזות של s. (המשלים של המחרוזת 0100 הוא לדוגמא 1011).

<u>שאלה 6 (14 נקודות)</u>

יהא A מערך בעל n מספרים. איבר x יקרא איבר רוב ב-A אם הוא מופיע <u>יותר</u> מ-n/2 פעמים. לדוגמא, האיבר 3 הוא איבר רוב במערך [3,3,2,3,5]. במערך [1,2,1,4] אין איבר רוב.

א. תאר/י, במודל ההשוואות, אלגוריתם דטרמיניסטי יעיל ככל האפשר שמקבל מערך A בגודל n ומחזיר את איבר הרוב שלו, אם קיים כזה. (במידה ואין איבר רוב, על האלגוריתם לדווח על כך.) הוכח/י בקצרה את נכונות האלגוריתם ונתח/י את סיבוכיותו.

בשני הסעיפים הבאים נניח שההשוואות היחידות שניתן לבצע על איברי המערך A הן השוואות מהצורה x=y שיחזירו 'כן' אם שני האיברים שווים, ו'לא' אחרת. (במידה ו-x≠y השוואה כזאת <u>לא</u> אומרת לנו מי גדול ממי.) להשוואות כאלה נקרא **<u>השוואות מסוג =</u>**.

ב. תאר/י, אלגוריתם דטרמיניסטי שסיבוכיותו $O(n \log n)$, שמשתמש רק בהשוואות מסוג = , שמקבל מערך A בגודל n ומחזיר את איבר הרוב שלו, אם קיים כזה. (במידה ואין איבר רוב, על האלגוריתם לדווח על כך.) הוכח/י בקצרה את נכונות האלגוריתם ונתח/י את סיבוכיותו.

ג. תאר/י, אלגוריתם **<u>הסתברותי</u>** יעיל ככל האפשר, שמשתמש רק בהשוואות מסוג = , שמקבל מערך A בגודל n **<u>שיש</u>** בו איבר רוב ומחזיר **<u>תמיד</u>** את איבר הרוב הזה. הוכח/י בקצרה את נכונות האלגוריתם ונתח/י את סיבוכיותו.

שאלה 7 (14 נקודות)

א. הגדר/י משפחה אוניברסלית H של פונקציות hash מקבוצה U אל $\{0,1,\ldots,m-1\}$.

ב. תהא T טבלת hash בעלת m תאים. לטיפול בהתנגשויות משתמשים ב-chaining. ממפים קבוצת S בעלת n איברים ל-A ע"י פונקצית hash h שנבחרת באקראי ממשפחה אוניברסלית של פונקציות hash. מהו n אם נתון שתוחלת מספר ההתנגשויות ב-T היא $m^{1/2}$? הוכח/י את תשובתך. (כל זוג איברים $x \neq y \in S$ שעבורו $h(x)=h(y)$ מגדיר התנגשות.)

ג. בהמשך לב', נניח שתחת פונקצית ה-hash המסוימת שנבחרה אכן היו לכל $m^{1/2}$ התנגשויות. כמה תאים בטבלת ה-hash יכולים להכיל לפחות $m^{1/8}$ איברים? הוכח/י את תשובתך.

בהצלחה !!!

מספר תעודת זהות:

מספר מחברת:

מועד ג׳

מספר קורס: 0368.2158

סמסטר ב׳, מועד ג׳, תשס״ז

28/12/2007

מבחן במבני נתונים

פרופ׳ חיים קפלן, אלעד ורבין, ליאור שפירא

משך המבחן שלוש וחצי שעות**(לא תינתן הארכה)**

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

הוראות כלליות:

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
4. **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.
5. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות. **תשובה ללא נמוק במקום שנדרש נמוק לא תזכה בנקודות.**
6. סיבוכיות זמני ריצה יש לכתוב כחסם הדוק ככל שתוכלו, בכתיב O().

בהצלחה!!

שאלה	ציון	ניקוד מקס׳
1		20
2		16
3		16
4		16
5		18
6		14
סה״כ		100

שאלה 1 (20 נקודות)

ענו על השאלות הבאות בקצרה (משפט או שניים לכל היותר)

א. יהיה X צומת המכיל מספר k בעץ חיפוש בינארי **מלא**, בו בכל צומת יש מספר. בהינתן כי k+1 נמצא בעץ ואין בעץ מספר y כך ש-k<y<k+1 היכן הצומת שמכיל את k+1?

ב. באיזו סיבוכיות זמן ניתן לבנות ערמה בעלת n צמתים?

ג. קוטר עץ T הינו המסלול הארוך ביותר בין שני צמתים **כלשהם**. תנו חסם עליון על קוטר עץ אדום-שחור בעל n צמתים.

O(   )

נמקו:

ד. נתון צומת v בעץ אדום-שחור. P1 ו-P2 הם אורך המסלול הקצר ביותר מ-v לעלה ואורך המסלול הארוך ביותר מ-v לעלה בהתאמה. תנו חסם תחתון על היחס בין P1 ל-P2. נמקו את תשובתכם.

מספר תעודת זהות:

מספר מחברת:

ה. באחד ממבני הנתונים לבעיית ה-union find השתמשנו בשיטה הנקראת ׳כיווץ מסלולים׳ (path compression), תארו את המבנה והשיטה בקצרה.

ו. בהינתן עץ T בערמת פיבונצ׳י, כך שלשורש העץ יש דרגה k. תנו חסם תחתון על מס׳ הצמתים בעץ, וחסם עליון על k

מס צמתים $= \Omega(\qquad)$

k $= O(\qquad)$

נמקו:

שאלה 2 (16 נקודות)

<u>חלק א'</u>

נתון עץ אדום שחור T (עם הערכים בעלים ועם מצביע מכל צומת לאביו). מבצעים m פעולות הכנסה לעץ כאשר בכל פעולה נתון עלה שלפניו או אחריו יש להכניס את האלמנט החדש.

הוכיחו כי עלות סדרת הפעולות הינה O(m) (ראו נספח עם חמשת המקרים של insert לעץ אדום שחור).

<u>חלק ב'</u>

עץ AVL הינו עץ חיפוש בינארי מאוזן המוגדר כך שעבור כל צומת T בעלת תת עץ ימני T_R ותת עץ שמאלי T_L, מתקיים כי: $|H(T_R)-H(T_L)|<=1$ כאשר H פ' גובה של תת עץ.

הוכיחו כי גובהו של עץ AVL בעל n מפתחות הוא O(logn).

שאלה 3 (16 נקודות)

נתונה המחרוזת S: BANANA

א. ציירו את ה-suffix tree של S.

ב. תארו אלגוריתם הרץ בזמן ליניארי המקבל כקלט מחרוזת S ומוצא את תת המחרוזת הקצרה ביותר החוזרת פעם אחת בלבד ב S. לדוגמה במחרוזת agtgga**c**atggg תת המחרוזת "c" הינה התשובה הנכונה.

ג. נתונות אותיות ותדירות הופעתן בטקסט. ציירו את עץ הופמן המתאים לקלט:

אות	א	ב	ג	ד	ה	ו	ז
תדירות	40	60	25	50	1	12	5

שאלה 4 (16 נקודות)

נתונה טבלת hash עם $m=11$ כניסות. כמו כן נתונות שתי פונקציות:

$h_1(key)=key \bmod m$

$h_2(key)=\{key \bmod (m-1)\} + 1$

א. הכניסו את המפתחות {11,3,7,22,18,33} לטבלה לפי הסדר (משמאל לימין) בכל אחת מהשיטות הבאות:

א.1. Linear probing כאשר התא ה-i שנבדוק למפתח k הוא $h(k,i)=(h_1(k)+i) \bmod m$

א.2. Double hashing כאשר h_1 פונקצית ה-hash ו-h_2 פונקצית המרווח. כלומר התא ה-i שנבדוק למפתח k הוא $h(k,i)=(h_1(k)+i*h_2(k)) \bmod m$

	Linear Probing	Double Hashing
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

ב. מה יקרה אם באו2. נשתמש ב-h_2 כפונקצית ה-hash וב-h_1 כפונקצית המרווח? כלומר
$h(k,i)=(h_2(k)+i*h_1(k)) \bmod m$

ג. הסבר מדוע כאשר $GCD(h_{step}(k),m)=1$ עבור כלk, התופעה בסעיף ב׳ אינה יכולה להתרחש. הפונקציה $h_{step}()$ היא פונקצית המרווח.

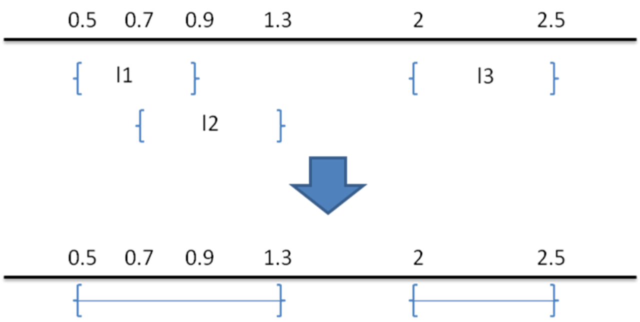
<u>שאלה 5 (18 נקודות)</u>

א. בהינתן קבוצה של n מספרים שונים ומספר k (כאשר n מתחלק ב-k ללא שארית), תארו אלגוריתם יעיל ככל שתוכלו, שמחלק את הקבוצה ל-k קבוצות שוות בגודלן $S_1,\ldots,S_k$ כך שכל המספרים ב-S_i קטנים מכל המספרים ב-S_{i+1} לכל $1 \le i \le k-1$. מהו זמן הריצה של האלגוריתם שתיארתם? נמקו.

לדוגמה: עבור קבוצת המספרים $\{1,3,6,7,10,12,15,16,20\}$ ו-$k=3$, נחלק לקבוצות הבאות: $\{1,3,6\}$ $\{7,10,12\}$, $\{15,16,20\}$.

ב. בהינתן n מספרים מתחום סדור, ומס' k כך ש-$k<n$, תאר אלגוריתם המוצא בזמן $O(n)$ את k המספרים הקרובים ביותר בערכם לחציון?

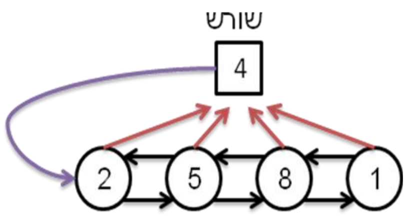
ג. נתונים n קטעים על הישר. תארו אלגוריתם המוצא מהו אורך איחוד הקטעים. לדוגמה עבור הקטעים $[0.5,0.9]$, $[0.7,1.3]$, $[2,2.5]$ נקבל:



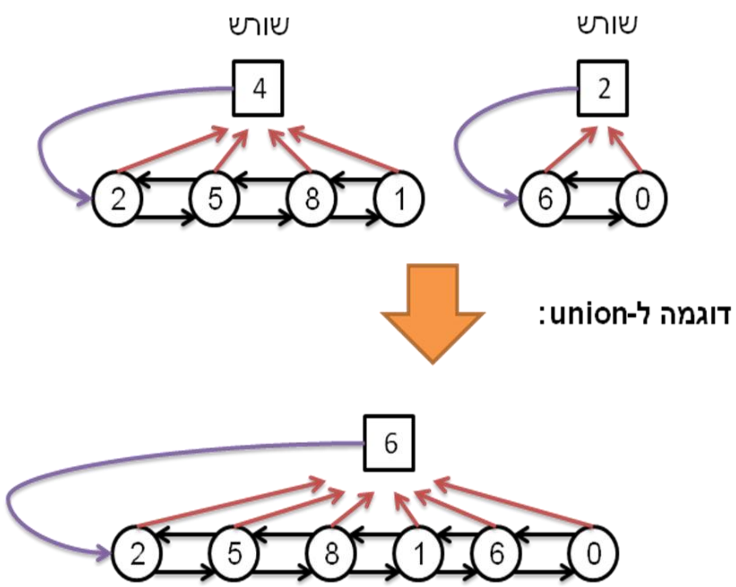
אורך איחוד הקטעים: 0.8+0.5=1.3

שאלה 6 (14 נקודות)

להלן מבנה נתונים הפותר את בעיית הunion find-. נחזיק קבוצה ע"י רשימה משורשרת דו-כיוונית של אבריה כאשר כל אלמנט מצביע לצומת מיוחד המייצג את הקבוצה (נכנה אותו שורש הקבוצה). שורש הקבוצה מצביע לרשימה המשורשרת וגם מכיל את מספר האלמנטים בקבוצה. לדוגמה הנה ייצוג של הקבוצה {2,5,8,1}, גודל הקבוצה (4) נשמר בשורש.



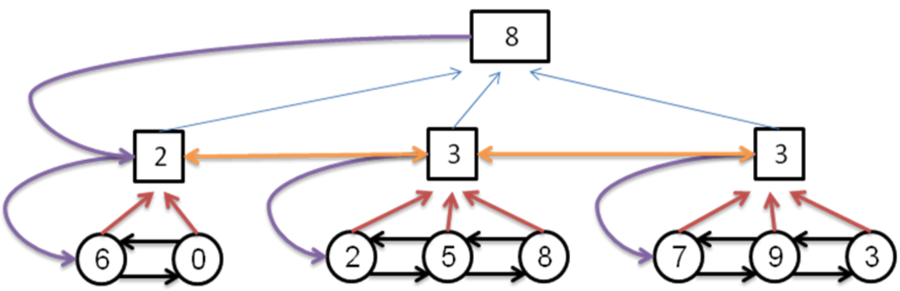
- פעולת ה-find מקבלת מצביע לאלמנט ומחזירה את המצביע לשורש שמאוכסן בצומת המייצג את האלמנט.
- פעולת ה-union מוסיפה את אברי הקבוצה הקטנה בזה אחר זה לרשימה של הקבוצה הגדולה ומעדכנת את המצביע לשורש באברים אלו, שיצביע לשורש בקבוצה הגדולה. בנוסף, הפעולה מעדכנת את מספר האברים בשורש הקבוצה הגדולה, ומוחקת את שורש הקבוצה הקטנה.



א. כמה זמן במקרה הגרוע ייקח לבצע סדרה בת m פעולות כאשר מתחילים מאוסף של n קבוצות, כל אחת מכילה רק אבר אחד? הנח ש-m>n. הוכח את נכונות תשובתך.

ב. נשנה את מבנה הנתונים באופן הבא. נסמן $k=\lfloor \log n \rfloor$. קבוצות שגודלן k לכל היותר נייצג כמו קודם. קבוצה S בגודל גדול מ-k תיוצג ע"י אוסף של תת קבוצות בגודל k בדיוק (נקרא להן **מלאות**), וקבוצה אחת לכל היותר בגודל קטן מ-k (**חסרה**). תתי קבוצות אלו מיוצגות כמו שתואר בחלק א' של השאלה. שורשי תת הקבוצות מסודרים ברשימה משורשרת דו-כיוונית ומצביעים לצומת חדש המייצג את הקבוצה כולה. שורש זה מכיל מצביע לרשימת שורשי תתי הקבוצות ואת גודל הקבוצה. הקבוצה החסרה (אם קיימת) תהיה תמיד ראשונה ברשימה.

דוגמה לקבוצה גדולה עם שתי קב' מלאות ואחת חסרה, k=logn=3



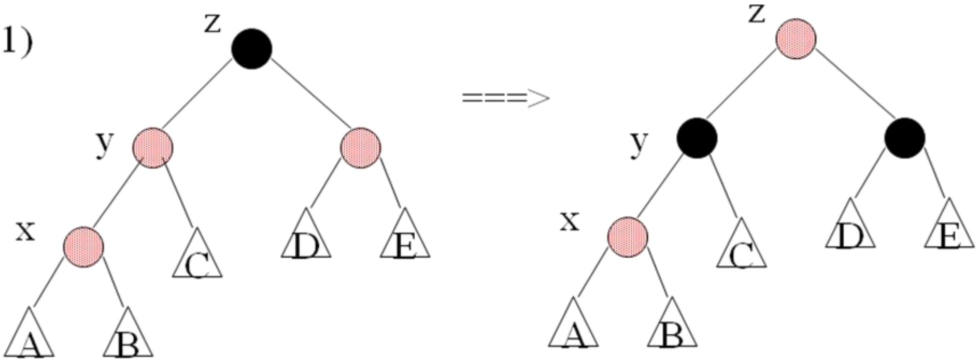
נבצע את הפעולות באופן הבא:

- נבצע **find** ע"י מציאת שורש תת הקבוצה המאוכסן בצומת המייצג את האיבר ומשם נמשיך לשורש הקבוצה (אם היא קב' גדולה).
- בכדי לבצע פעולת **union** של תת קבוצות, נשתמש בפעולת ה-union כמו שהוגדרה קודם, עם השינוי הבא: במידה וגודל שתי התת קבוצות יחדיו הוא יותר מ-k אזי ה-union יחזיר שתי תת קבוצות, אחת מלאה ואחת חסרה. union זה מתבצע כמו קודם רק שאין הוא מעתיק את כל איברי הקבוצה הקטנה במידה ואין יותר מקום בקבוצה הגדולה.
- **union** של קבוצות נבצע באופן הבא:
  - אם שתי הקבוצות קטנות וגודלן יחדיו קטן או שווה ל-k אזי הפעולה תבוצע כאיחוד תתי קבוצות כמו קודם.
  - אם שתי הקבוצות קטנות וגודלן יחדיו גדול מ-k אזי נאחד את תתי הקבוצות ונקבל תת קבוצה אחת מלאה ואחת חסרה. ניצור קבוצה חדשה שאלו הן תתי הקבוצות שלה ונחזיר אותה.
  - אם קב' אחת קטנה ואחת גדולה, נאחד את הקבוצה הקטנה עם תת הקבוצה החסרה של הקבוצה הגדולה. נוסיף את תתי הקבוצות המתקבלות לרשימת תתי הקבוצות של הקבוצה הגדולה. נחזיר את הקבוצה הגדולה.
  - אם שתי הקבוצות גדולות נעביר את המלאות מן הקבוצה הקטנה יותר לרשימת תת הקבוצות בקב' הגדולה יותר. כמו כן נאחד את החסרות ונוסיף את תתי הקבוצות המתקבלות לרשימת תתי הקבוצות של הקבוצה הגדולה. נחזיר את הקבוצה הגדולה.

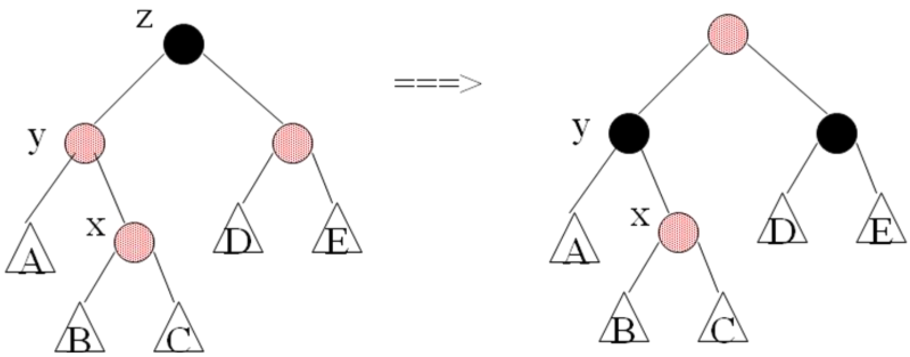
הוכח כי זמן הריצה של סדרה בת m פעולות כאשר מתחילים מאוסף של n קבוצות, כל אחת מכילה רק אבר אחד הוא $O(n\log\log(n) + m)$.

<u>נספח לשאלה 2 : חמשת מקרי ה-insert של עץ אדום שחור</u>

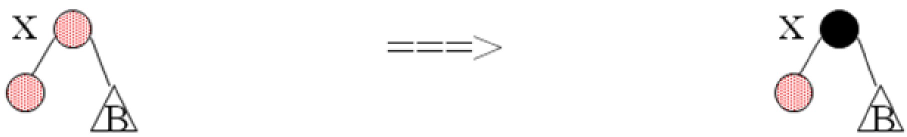
(1)



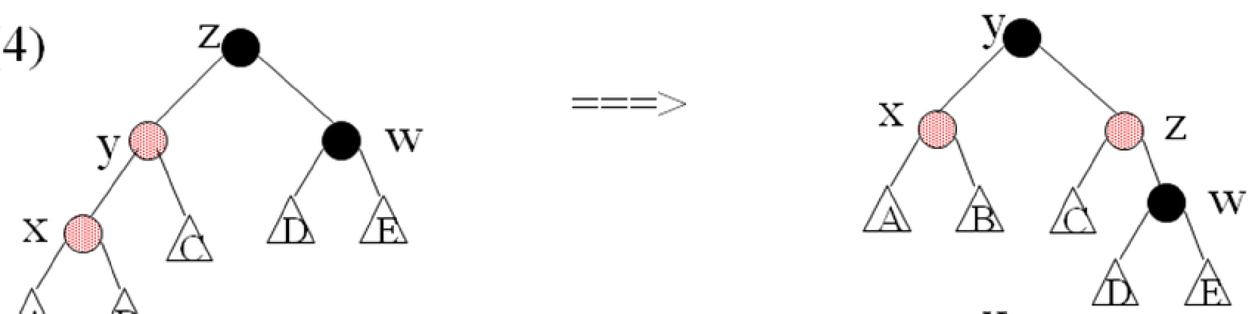
(2)



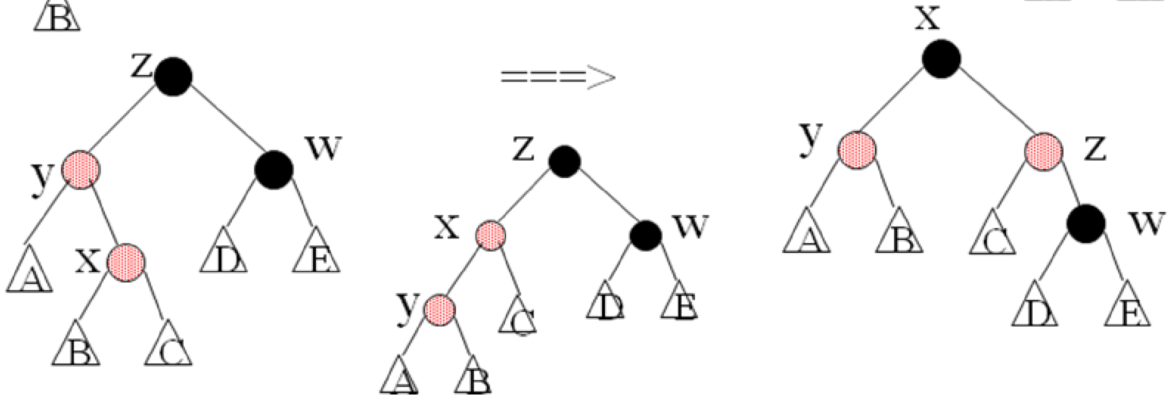
(3)



(4)



(5)



מספר קורס: 0368.2158

סמסטר ב', מועד ב', תשס"ז
י"ד באב תשס"ז 18/10/2007

מבחן במבני נתונים

פרופ' חיים קפלן, אלעד ורבין, ליאור שפירא

משך המבחן שלוש וחצי שעות **(לא תינתן הארכה)**
הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

הוראות כלליות:

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
4. **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.
5. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות. **תשובה ללא נמוק במקום שנדרש נמוק לא תזכה בנקודות.**
6. סיבוכיות זמני ריצה יש לכתוב כחסם הדוק ככל שתוכלו, בכתיב O().

בהצלחה!!

שאלה	ציון	ניקוד מקס'
1		20
2		14
3		18
4		20
5		14
6		14
סה"כ		100

<u>שאלה 1 (20 נקודות)</u>

א. נתון מערך בגודלn שבו כל תא מכיל את התו '+' או התו '-'. ידוע כי האיבר הראשון במערך שווה ל-'+' והאחרון שווה ל-'-'. תאר אלגוריתם שרץ בזמן **O(log n)** ומוצא אינדקס j כך ש-A[j]='+' וגם A[j+1]='-'.

דוגמה:

	+	+	_	+	+	+	_	_	+	_
index	0	1	2	3	4	5	6	7	8	9

בדוגמה שלהלן התשובות החוקיות הןj=1, j=5 ו-j=8. (מותר להחזיר איזו מהן שרוצים. לא צריך להחזיר את כולן).

פתרון:

ב. הראה חסם תחתון שמוכיח שלכל אלגוריתם שפותר את השאלה מסעיף א', יש סיבוכיות **Ω(log n)** ב-worst case.

(שים לב שנדרשת זהירות בטיעון בשל העובדה שעבור קלטים מסויימים יש יותר מתשובה אחת חוקית).

פתרון:

ג. נתון מערך בגודל n המכיל מספרים חיוביים. כל האיברים במערך שונים. כמו כן ידוע שהאיבר הראשון והאחרון במערך הם האיברים הקטנים ביותר. תאר אלגוריתם שרץ בזמן **O(log n)** ומוצא אינדקס j כך ש-$A[j-1]<A[j]$ וגם $A[j+1]<A[j]$. במקרה שיש יותר מתשובה אחת חוקית, ניתן להחזיר את איזו מהן שרוצים.

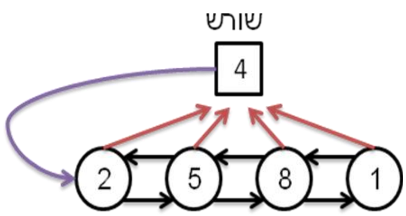
	1	3	4	5	6	7	26	15	14	2
index	0	1	2	3	4	5	6	7	8	9

בדוגמה שלהלן התשובה החוקית היא $j=6$.

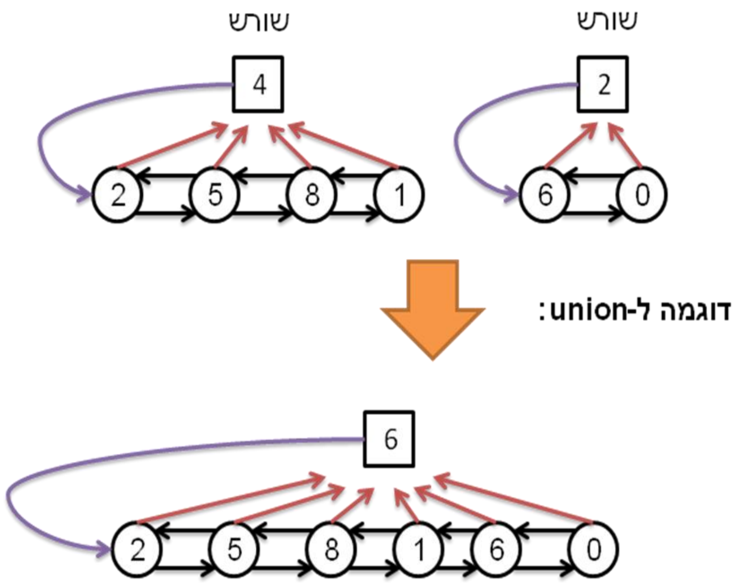
פתרון:

שאלה 2 (14 נקודות)

להלן מבנה נתונים הפותר את בעיית הunion find. נחזיק קבוצה ע"י רשימה משורשרת דו-כיוונית של אבריה כאשר כל אלמנט מצביע לצומת מיוחד המייצג את הקבוצה (נכנה אותו שורש הקבוצה). שורש הקבוצה מצביע לרשימה המשורשרת וגם מכיל את מספר האלמנטים בקבוצה. לדוגמה הנה ייצוג של הקבוצה {2,5,8,1}, גודל הקבוצה (4) נשמר בשורש.



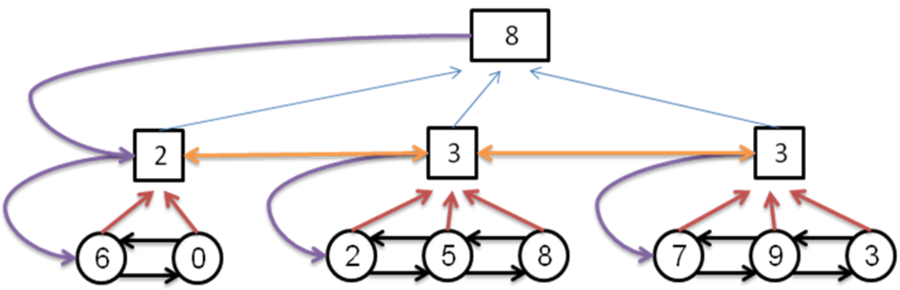
- פעולת ה-find מקבלת מצביע לאלמנט ומחזירה את המצביע לשורש שמאוכסן בצומת המייצג את האלמנט.
- פעולת ה-union מוסיפה את אברי הקבוצה הקטנה בזה אחר זה לרשימה של הקבוצה הגדולה ומעדכנת את המצביע לשורש באברים אלו, שיצביע לשורש בקבוצה הגדולה. בנוסף, הפעולה מעדכנת את מספר האברים בשורש הקבוצה הגדולה, ומוחקת את שורש הקבוצה הקטנה.



א. כמה זמן במקרה הגרוע ייקח לבצע סדרה בת m פעולות כאשר מתחילים מאוסף של n קבוצות, כל אחת מכילה רק איבר אחד? הנח ש-m>n. הוכח את נכונות תשובתך.

ב. נשנה את מבנה הנתונים באופן הבא. נסמן $k=\lfloor \log n \rfloor$. קבוצות שגודלן k לכל היותר נייצג כמו קודם. קבוצה S בגודל גדול מ-k תיוצג ע"י אוסף של תת קבוצות בגודל k בדיוק (נקרא להן **מלאות**), וקבוצה אחת לכל היותר בגודל קטן מ-k (**חסרה**). תתי קבוצות אלו מיוצגות כמו שתואר בחלק א' של השאלה. שורשי תת הקבוצות מסודרים ברשימה משורשרת דו-כיוונית ומצביעים לצומת חדש המייצג את הקבוצה כולה. שורש זה מכיל מצביע לרשימת השורשים ואת גודל הקבוצה. הקבוצה החסרה (אם קיימת) תהיה תמיד ראשונה ברשימה.

דוגמה לקבוצה גדולה עם שתי קב' מלאות ואחת חסרה, k=logn=3



נבצע את הפעולות באופן הבא:

- נבצע **find** ע"י מציאת שורש תת הקבוצה המאוכסן בצומת המייצג את האיבר ומשם נמשיך לשורש הקבוצה (אם היא קב' גדולה).
- בכדי לבצע פעולת **union** על תת קבוצות, נשתמש בפעולת ה-union כמו שהוגדרה קודם, עם השינוי הבא: במידה וגודל שתי התת קבוצות יחדיו הוא יותר מ-k אזי ה-union יחזיר שתי תת קבוצות, אחת מלאה ואחת חסרה. union זה מתבצע כמו קודם רק שאין הוא מעתיק את כל איברי הקבוצה הקטנה במידה ואין יותר מקום בקבוצה הגדולה.
- **union** של קבוצות נבצע באופן הבא:
    - אם שתי הקבוצות קטנות וגודלן יחדיו קטן או שווה ל-k אזי הפעולה תבוצע כאיחוד תתי קבוצות כמו קודם
    - אם שתי הקבוצות קטנות וגודלן יחדיו גדול מ-k אזי נאחד את תתי הקבוצות ונקבל תת קבוצה אחת מלאה ואחת חסרה. ניצור קבוצה חדשה שאלו הן תתי הקבוצות שלה ונחזיר אותה.
    - אם קב' אחת קטנה ואחת גדולה, נאחד את הקבוצה הקטנה עם תת הקבוצה החסרה של הקבוצה הגדולה. נוסיף את תתי הקבוצות המתקבלות לרשימת תתי הקבוצות של הקבוצה הגדולה. נחזיר את הקבוצה הגדולה.
    - אם שתי הקבוצות גדולות נעביר את המלאות מן הקבוצה הקטנה יותר לרשימת תת הקבוצות בקב' הגדולה יותר. כמו כן נאחד את החסרות ונוסיף את תתי הקבוצות המתקבלות לרשימת תתי הקבוצות של הקבוצה הגדולה. נחזיר את הקבוצה הגדולה.

הוכח כי זמן הריצה של סדרה בת m פעולות כאשר מתחילים מאוסף של n קבוצות, כל אחת מכילה רק אבר אחד הוא $O(n\log\log(n) + m)$.

שאלה 3 (18 נקודות)

נתונה המחרוזת S: ACACACCA

א. ציירו את ה-suffix tree של S.

ב. נגדיר תת מחרוזת z באורך k להיות חזרה מכסימלית ימנית במחרוזת S אם קיימים i ו-j, $i \neq j$ כך ש:

$$Z = S_i S_{i+1} \ldots S_{i+k-1}$$

$$Z = S_j S_{j+1} \ldots S_{j+k-1}$$

$$S_{i+k} \neq S_{j+k}$$

תאר אלגוריתם לינארי המקבל כקלט מחרוזת S ומספר k ומחשב את כל החזרות המכסימליות הימניות באורך k. עבור כל חזרה האלגוריתם מדווח את האינדקס בו מתחיל המופע הראשון שלה.

ג. נתונות אותיות ותדירות הופעתן בטקסט. ציירו את עץ הופמן המתאים לקלט:

אות	א	ב	ג	ד	ה	ו	ז
תדירות	40	100	25	5	100	70	15

שאלה 4 (20 נקודות)

נתונה טבלת hash עם $m=11$ כניסות. כמו כן נתונות שתי פונקציות:

$h_1(key)=key \bmod m$

$h_2(key)=\{key \bmod (m-1)\} + 1$

א. הכניסו את המפתחות {22,1,13,11,24,33,18 } לטבלה לפי הסדר (משמאל לימין) בכל אחת מהשיטות הבאות:

א.1. Linear probing כאשר התא ה-i שנבדוק למפתח k הוא $h(k,i)=(h_1(k)+i) \bmod m$

א.2. Double hashing כאשר h_1 פונקצית ה-hash ו-h_2 פונקצית המרווח. כלומר התא ה-i שנבדוק למפתח k הוא $h(k,i)=(h_1(k)+i*h_2(k)) \bmod m$

	Linear Probing	Double Hashing
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

ב. מה יקרה אם באג.2 נשתמש ב-h_2 כפונקצית ה-hash וב-h_1 כפונקצית המרווח? כלומר

$h(k,i)=(h_2(k)+i*h_1(k)) \bmod m$

ג. הסבר מדוע כאשר $GCD(h_{step}(k),m)=1$ עבור כל k, התופעה בסעיף ב' אינה יכולה להתרחש. הפונקציה $h_{step}()$ היא פונקצית המרווח.

שאלה 5 (14 נקודות)

הוכיחו כי לא יכול להיות אלגוריתם במודל ההשוואות בו עבור קלט באורך n, לפחות חצי מהפרמוטציות האפשריות של המספרים מ-1 עד n ניתנות למיון בזמן ליניארי.
(עבור המחצית השניה של הפרמוטציות, לאלגוריתם מותר להחזיר כל דבר, אפילו סדר שגוי).

שאלה 6 (14 נקודות)

ענו על שני הסעיפים הבאים

א. תארו מבנה נתונים התומך בפעולות הבאות על קבוצה S מתחום סדור מלא

- Quarternian(S): אם יש כרגע ב-S n איברים, מחזיר את ה-$\lceil n/4 \rceil$-י בגודלו
- Average(S): מחזיר את ממוצע האיברים ב-S
- Min(S): מחזיר את האבר הקטן ביותר ב-S
- Max(S): מחזיר את האבר הגדול ביותר ב-S
- Insert(x,S): מוסיף אבר x ל-S
- Delete(x,S) מניחים ש-x נמצא לפני הפעולה ב-S. הפעולה מוציאה את x מ-S

על הפעולות quarternian,average,min,max לקחת O(1) זמן במקרה הגרוע, ועל הפעולות insert ו-delete לקחת O(logn) זמן במקרה הגרוע. כמו כן תארו בקצרה כיצד לבצע כל פעולה.

ב. שנו במידת הצורך את מבנה הנתונים שהגדרתם בסעיף הקודם כך שנוכל גם למצוא את האלמנט ה-$i+\lceil n/4 \rceil$ בגודלו בזמן $O(\log(i))$. תארו במדויק כיצד לבצע פעולה זו.

מספר תעודת זהות: ____ 1 מתוך 1 מועד א', גירסה 1

מספר מחברת: ____

מספר קורס: 0368.2158

סמסטר ב', מועד א', תשס"ז

י"ד באב תשס"ז 29/7/2007

מבחן במבני נתונים + תשובות

פרופ' חיים קפלן, אלעד ורבין, ליאור שפירא

משך המבחן שלוש וחצי שעות **(לא תינתן הארכה)**

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
4. <u>**אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.</u>
5. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות. **<u>תשובה ללא נמוק במקום שנדרש נמוק לא תזכה בנקודות.</u>**
6. סיבוכיות זמני ריצה יש לכתוב כחסם הדוק ככל שתוכלו, בכתיב O( ).

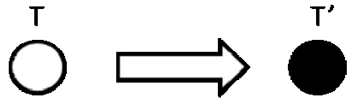
בהצלחה!!

שאלה	ציון	ניקוד מקס'
1		20
2		12
3		12
4		15
5		8
6		13
7		10
8		10
סה"כ		100

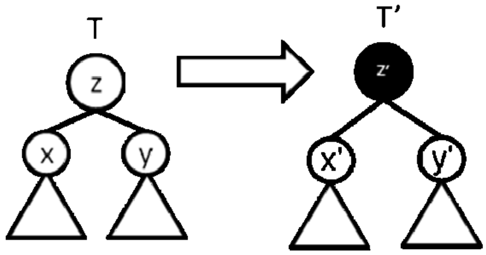
<u>שאלה 1 (20 נקודות)</u>

יהי T עץ 2-4+ (האלמנטים נמצאים בעלים, בצמתים הפנימיים יש רק מפתחות להנחות את החיפוש). נהפוך את T לעץ אדום שחור 'T ע"פ אחד המקרים הבאים:

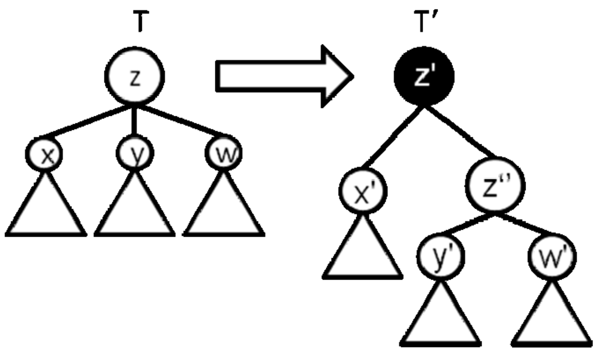
1. במידה ו-T מכיל עלה יחיד, גם 'T יכיל עלה יחיד שחור ובו אותו אלמנט.



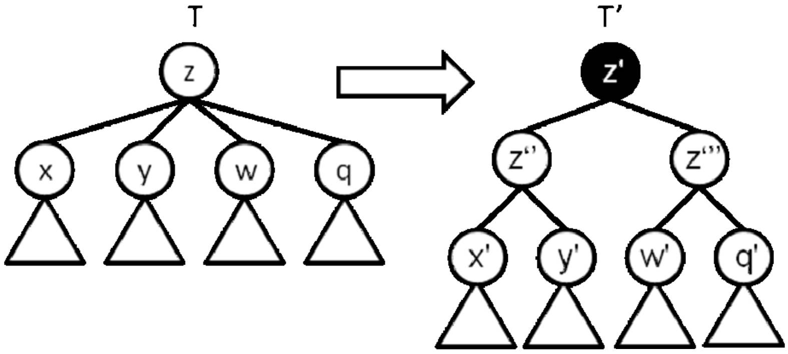
2. במידה והשורש של T הוא צומת z עם 2 בנים x,y אזי שורשו של 'T יהיה צומת שחור 'z. הבן השמאלי של 'z יהיה שורש העץ 'x שמתקבל מהפעלת התהליך רקורסיבית על תת העץ ששורשו x, ואילו הבן הימני של 'z יהיה שורש העץ 'y שמתקבל מהפעלת התהליך רקורסיבית על תת העץ ששורשו y. המפתח בצומת 'z זהה למפתח ב-z.



3. במידה והשורש z של T הוא צומת עם 3 בנים x,y,w אזי שורשו של 'T יהיה צומת שחור 'z. הבן השמאלי של 'z יהיה שורש העץ 'x שמתקבל מהפעלת התהליך רקורסיבית על תת העץ ששורשו x. הבן הימני של 'z יהיה צומת אדום "z. לצומת "z יש שני בנים, השמאלי יהיה שורש העץ 'y שמתקבל מהפעלת התהליך רקורסיבית על תת העץ ששורשו y ואילו הימני יהיה שורש העץ 'w שמתקבל מהפעלת התהליך רקורסיבית על תת העץ ששורשו w. המפתחות ב-'z ו-"z הם המפתחות מ-z.



4. במידה והשורש של T הוא צומת עם 4 בנים x,y,w,q אזי שורשו של T' יהיה צומת שחור z' ולו שני בנים אדומים z'',z'''. בניו של z'' יתקבלו מהפעלת התהליך רקורסיבית על תתי העצים ששורשם x,y ואילו בניו של z''' יתקבלו ע״י הפעלת התהליך רקורסיבית על תתי העצים ששורשם w,q. המפתחות ב-z',z'',z''' הם המפתחות מ-z (השורש של T).

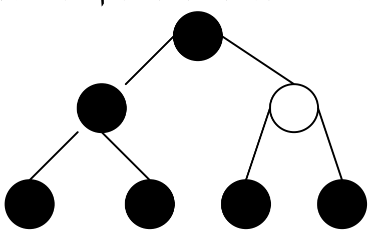


ענה על השאלות הבאות:

א. **הוכיחו** כי 'T הינו בהכרח עץ אדום-שחור חוקי לכל עץ +2-4 T הניתן כקלט.

קל לראות שהעץ המתקבל הוא אכן עץ חיפוש, ושכל העלים שחורים. נשאר להוכיח שהכלל האדום והכלל השחור מתקיימים.
הוכחת הכלל השחור: קל להוכיח באינדוקציה שהעומק השחור של עלה בעץ האדום-שחור שווה לעומק של העלה המתאים לו בעץ 2-4. עתה, היות והעומק של כל העלים בעץ 2-4 שווה, אז העומק השחור של כל העלים בעץ אדום-שחור שווה, וזה מוכיח את הכלל השחור.
הכלל האדום מתקיים כי מהגדרת הפעולות קל לראות שלכל קודקוד אדום יש אב שחור, ולכן אין שני אדומים שהם אב ובן.

<u>טעות נפוצה</u>: חלק מהסטודנטים ניסו להוכיח שהכלל השחור מתקיים בעזרת אינדוקציה. לרוע המזל, הרבה סטודנטים השתמשו כאן באינדוקציה באופן שגוי. למשל, תת העץ הימני ותת העץ השמאלי של העץ הזה שניהם מקיימים את הכלל השחור, אך בכל זאת, העץ כולו לא מקיים את התנאי השחור. ההוכחות השגויות של הרבה סטודנטים לשאלה זו, יכולות להוכיח גם שהעץ הזה מקיים את הכלל השחור.



ב. האם כל עץ אדום שחור הוא תמונה של עץ +2-4 כלשהוא? **הוכיחו** אם כן, הביאו **דוגמה נגדית** אם לא.

התשובה היא לא. למשל לא ניתן לקבל עץ שבו שלושה קודקודים: שורש אדום, ושני בנים שחורים.

ג. הגדירו התאמה "הפוכה" המתרגמת עץ אדום-שחור לעץ +2-4.

יש שני פתרונות, כאשר הראשון הוא הפתרון אליו התכוונו, והשני הוא פתרון שגם הוא נכון, אבל לא עליו חשבנו. כמובן, על שניהם ניתן ניקוד מלא.

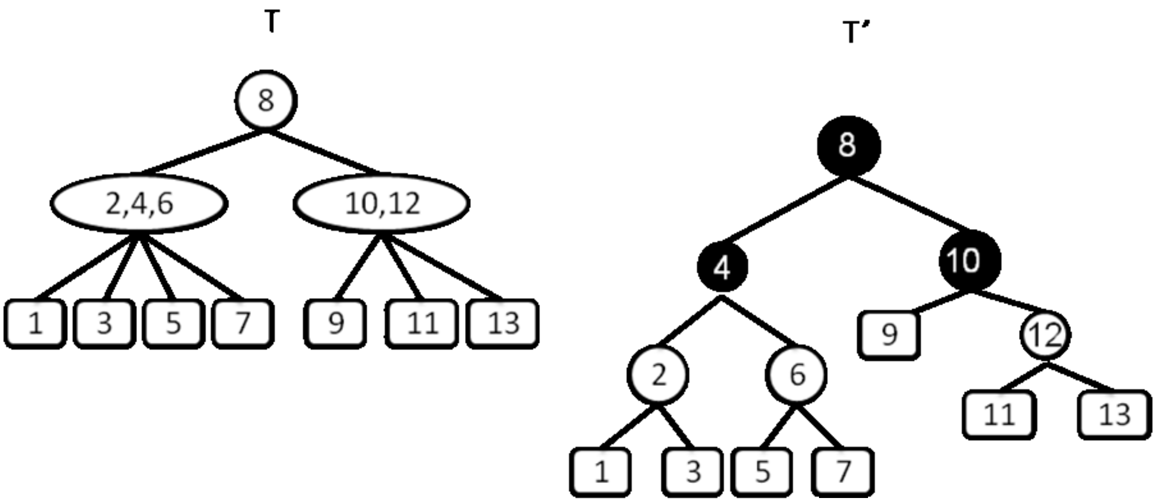
פתרון ראשון: ההתאמה עובדת כך: נבחר קודקוד אדום בעץ, x. נסמן את אביו ב-y. ניפטר מ-x ונהפוך את בניו לבנים ישירים של y. את המפתח שנמצא ב-x נעביר ל-y כך שיחצוץ בין שני בניו החדשים. נחזור על התהליך הזה לכל הקודקודים האדומים בעץ. המקרה שהשורש אדום דורש טיפול מיוחד: נאחד את שני בניו זה לזה, ונעיף את השורש האדום. נעביר את המפתח מהשורש האדום להיות המפתח האמצעי בשורש החדש שנוצר. קל לראות שהפעולה הזאת נותנת עץ 2-4 חוקי. (ההוכחה דומה לסעיף א'. לא נכלול אותה בפתרון).

<u>טעויות נפוצות:</u> הרבה סטודנטים לא טיפלו במקרה שבו השורש אדום. חלק מהסטודנטים פשוט הפכו את הגדרת הטרנספורמציה מ-2-4 לאדום-שחור שהוגדרה בשאלה, ולא טרחו לפרט, למשל, את ההפכי של המקרה הסימטרי לפעולה 3.

פתרון שני: קוראים את העץ האדום-שחור בעזרת קריאת in-order, ואז מהמערך שהתקבל, יוצרים עץ 2-4 כמו שראינו בתרגיל בית מס' 4 שאלה 2(b).

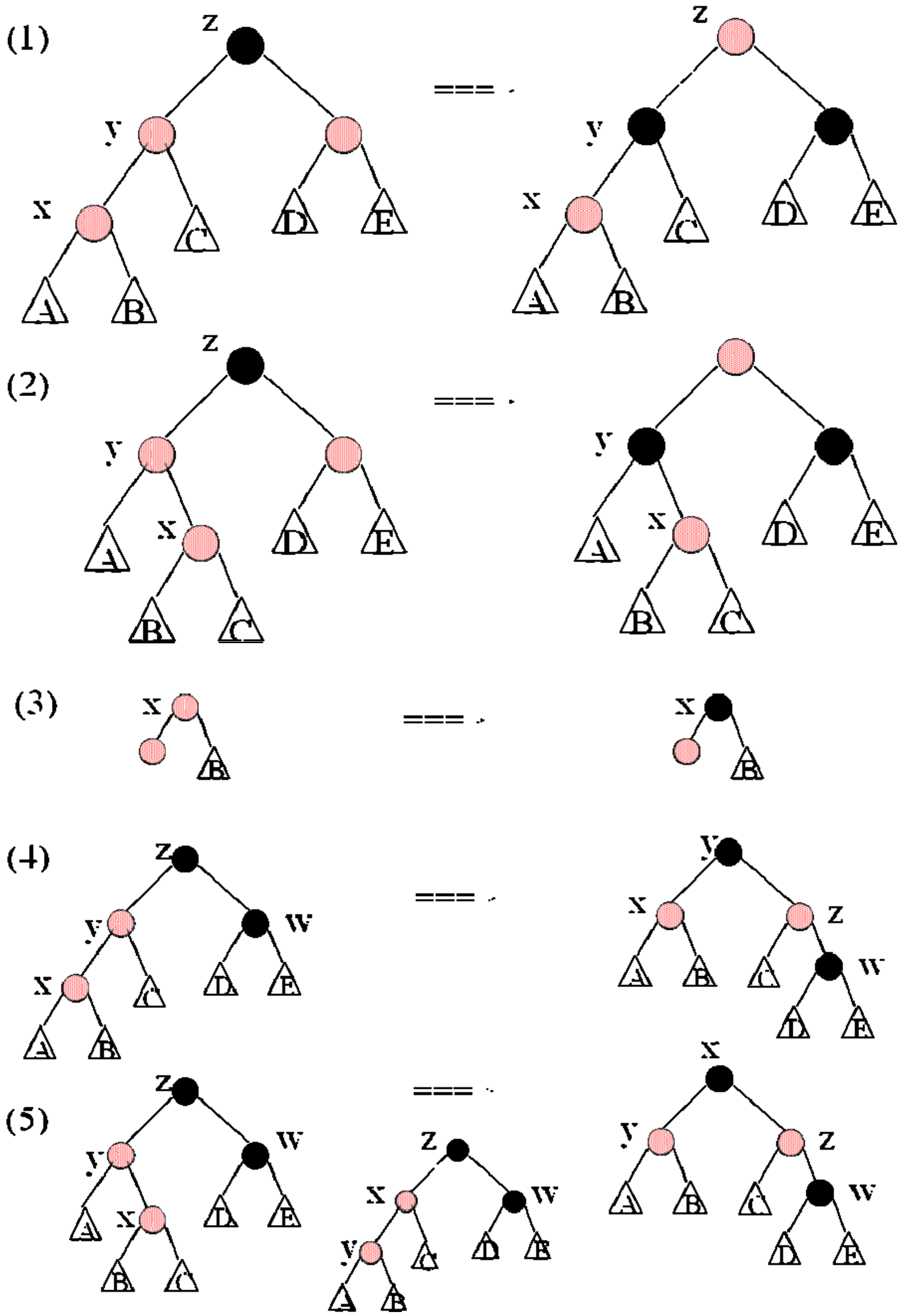
ד. לפניכם עץ 2-4+ T ועץ אדום-שחור 'T. 'T התקבל ע"י הפעלת התהליך המתואר בשאלה על T. נבצע פעולת insert(x) על T לקבלת עץ H. נבצע פעולת insert(x) על 'T לקבלת עץ 'H. האם לכל x 'H הוא העץ המתאים ל-H ע"פ ההתאמה שהוגדרה בשאלה? אם כן הוכיחו, אחרת תנו דוגמה לערך x עבורו 'H ו-H אינם תואמים .

<u>שימו לב לנספח בעמוד הבא ובו פירוט חמשת מקרי ה-insert של עץ אדום שחור!</u>



התשובה היא לא. אם מכניסים x=0, אז ב-H לשורש יהיו שלושה בנים, ובעץ האדום-שחור שמתקבל מ-H, בתת-העץ הימני של השורש יהיו יותר עלים מאשר בתת-העץ השמאלי של השורש. לעומת זאת, אם מכניסים את 0 ל-'T, אז יתבצע recoloring, ואז בתת-העץ השמאלי של השורש יהיו חמישה עלים, ובצד ימין רק שלושה.

<u>נספח לשאלה 1: חמשת מקרי ה-insert של עץ אדום שחור</u>



<u>שאלה 2 (12 נקודות)</u>

בשאלה זו נדון במונה בינארי בו עלות פעולה (הוספת או הפחתת 1) זהה למספר הביטים שאנו משנים תוך כדי ביצוע הפעולה. ראינו בכיתה כי אם נתחיל במצב בו המונה מאופס, ונבצע n פעולות increment (הוספה של 1), אזי העלות amortized לפעולה היא $O(1)$. עתה נניח כי אנו מבצעים סדרה של n פעולות increment ו-decrement (הורדת 1) כאשר מתחילים ממונה מאופס.

א. הראו שמבלי להגיע לערכים שליליים במונה, קיימת סדרה בת n פעולות increment ו-decrement (המתחילה ממונה 0), כך שהעלות amortized של פעולה היא $\Omega(\log n)$.

יהי k מספר בין n/2 ל-n/4 כל שבייצוג הבינארי שלו יש רק 1-ים. סדרת הפעולות תהיה k פעמים פעולות increment, ואז (n-k)/2 חזרות על שתי הפעולות increment,decrement. קל לראות שכל אחת מהפעולות שמבוצעות בזמן החזרות הללו לוקחת $\Omega(\log n)$, ולכן העלות הכוללת היא $\Omega(n \log n)$, כנדרש.

ב. בכדי לשפר את יעילות המונה נשתמש בספרות 0,+1,-1 (במקום רק ב-0 ו-1). ערכו של מספר המיוצג ע"י סדרת הספרות $t_{k-1},...,t_0$ מוגדר להיות:

$$\sum_{i=0}^{k-1} t_i 2^i$$

למשל **1- 0 1** הינו ייצוג של $3 = 2^2 - 2^0$.

פעולת increment של מספר בייצוג כזה מתבצעת באופן דומה לבצועה במערכת המספרים הרגילה. מוסיפים 1 לספרה הימנית ביותר, אם ערכה הפך ל-2 הוא משתנה ל-0 וגוררים את העודף לספרה הבאה משמאל. Decrement מתבצע בצורה דומה, מורידים 1 מהספרה הימנית ביותר, אם ערכה הפך ל-2- הופכים אותו ל-0 וגוררים את החוסר (1-) לספרה שמשמאל.
כמו קודם נגדיר את עלות הפעולה להיות מספר הספרות המשתנות כאשר מבצעים את הפעולה.
הוכיחו בצורה ברורה כי בייצוג שכזה העלות של סדרת n פעולות increment ו-decrement כאשר מתחילים ממונה שערכו 0 היא $O(n)$.

נגדיר פונקציית פוטנציאל להיות שווה למס' ה-1-ים וה-(1-)-ים. קל לראות שבכל פעולה, כל הביטים שמשתנים ישתנו ל-0, פרט אולי לביט אחד. לכן בכל פעולה, השינוי בפוטנציאל "משלם" על שינוי כל הביטים שמשתנים ל-0, ונשאר רק לשלם על עוד שינוי ביט אחד, וההשפעה שלו על הפוטנציאל, וזה שווה ל-$O(1)$.

<u>שאלה 3 (12 נקודות)</u>

ענו על שני הסעיפים הבאים

א. תארו מבנה נתונים התומך בפעולות הבאות על קבוצה S מתחום סדור מלא

- Median(S): מחזיר את החציון ב-S (אם מס' אברים זוגי – חציון תחתון)
- Min(S) : מחזיר את האבר הקטן ביותר ב-S
- Max(S) : מחזיר את האבר הגדול ביותר ב-S
- Insert(x,S) : מוסיף אבר x ל-S
- Delete(x,S) מניחים ש-x נמצא לפני הפעולה ב-S. הפעולה מוציאה את x מ-S

על הפעולות median,min,max לקחת O(1) זמן במקרה הגרוע, ועל הפעולות insert ו-delete לקחת O(logn) זמן במקרה הגרוע. כמו כן תארו בקצרה כיצד לבצע כל פעולה.

היו שני פתרונות נפוצים.

פתרון ראשון: שומרים median-heap כמו שלמדנו בכתה. כלומר, שתי ערימות: אחת ערימת-מקסימום ואחת ערימת מינימום, כאשר כל איבר בערימת המינימום גדול מכל איבר בערימת המקסימום, ובנוסף הגודל של ערימת המקסימום שווה או קטן בבדיוק 1 מהגודל של ערימת המינימום. בפעולת insert או delete מכניסים/מוציאים את האיבר מ-/ל- ערימה המתאימה, ואם צריך מעבירים איבר מערימה אחת לאחרת כדי לאזן את הגדלים. ה-median תמיד יהיה האיבר המינימלי בערימת המינימום. בנוסף, מחזיקים ערימת מינימום וערימת מקסימום שמחזיקות את כל האיברים, כדי לענות על שאילתות ה-min וה-max.

טעויות נפוצות: הרבה סטודנטים תיארו לא נכון את ה-median-heap.

פתרון שני: שומרים עץ אדום-שחור שמכיל שדות size בכל קודקוד, ושומרים בנוסף שלושה פוינטרים: אחד שמצביע על האיבר המינימלי, אחד על המקסימלי, ואחד על החציון. אז ברור שפעולות min,max,median פשוטות ומתבצעות בזמן קבוע. בפעולות insert ו-delete מכניסים או מוציאים כרגיל, ואז מוצאים את המינימום, מקסימום, וחציון מחדש בזמן לוגריצמי בעזרת שימוש בפעולת סדר סטטיסטי בעץ, ומעדכנים את המצביעים.

הערה: במקום לשמור שדות size, ניתן לשמור רשימה מקושרת על האלמנטים לפי גודל, או הרבה פתרונות אחרים.

ב. שנו במידת הצורך את מבנה הנתונים שהגדרתם בסעיף הקודם כך שנוכל גם למצוא את האלמנט ה- i בגודלו מעל החציון (כלומר ישנם i אלמנטים שגדולים מהחציון וקטנים או שווים לאבר זה) בזמן O(log(i)). תארו במדויק כיצד לבצע פעולה זו.

פתרון ראשון: מממשים median-heap כמו קודם, אבל במקום ה-min-heap שומרים עץ אדום-שחור עם יכולת לעשות finger-search (כלומר, הענף השמאלי ביותר מכיל גם מצביעים כלפי מעלה). היות והחציון הוא האיבר המינימלי בעץ, אז ה--finger search נותן בדיוק את מה שאנחנו צריכים.

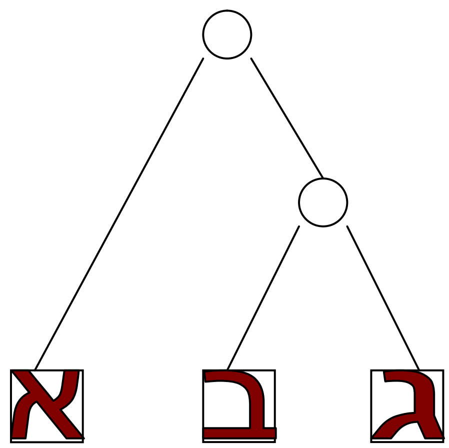
פתרון שני: אם רוצים להחזיק רק עץ אחד, צריך לעבוד קשה הרבה יותר. למשל, אפשר להחזיק בכל רמה רשימה מקושרת שמאפשרת לעבור ימינה ושמאלה בין כל הקודקודים ששייכים לאותה רמה, ואז לתאר אלגוריתם שהוא הכללה של finger search. הפתרון הראשון הרבה יותר מומלץ. כמעט אף סטודנט לא נתן פתרון נכון, מדוייק ושלם בשיטה הזאת

טעויות נפוצות: הרבה סטודנטים חשבו שעץ עם מצביעים דו-כיוניים (מאב לבניו וגם מבן לאביו) מספיק כדי לממש את הפתרון השני, אבל האמת היא שזה לא עובד. למשל, אם החציון הוא העלה הכי ימני בתת העץ השמאלי של השורש, אז כדי למצוא את האיבר הגדול ממנו ב-1, צריך לעלות עד השורש ובחזרה, וזו עלות של O(log n).
עוד שגיאה נפוצה היא שהרבה סטודנטים חשבו שהחציון יהיה בשורש של העת האדום-שחור. אין שום סיבה שזה בהכרח יקרה..

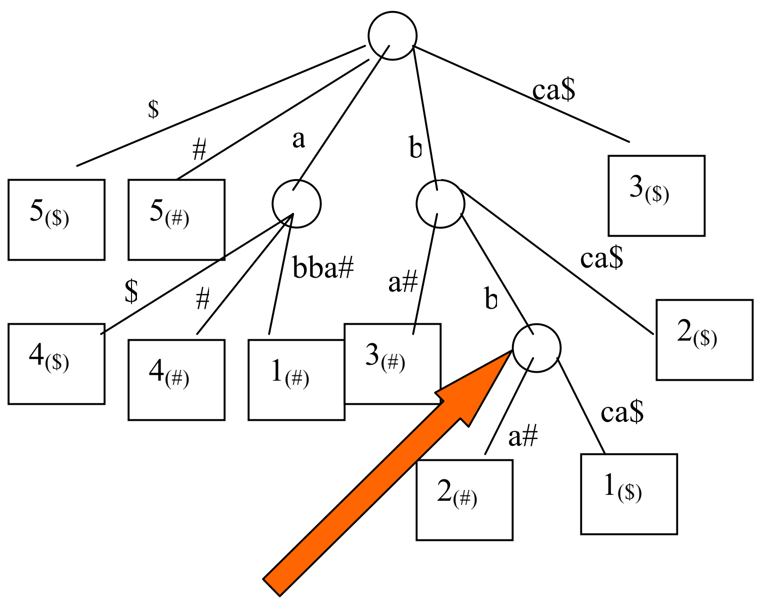
<u>שאלה 4 (15 נקודות)</u>

א. במחרוזת S מופעה האות א' מיליון פעמים, האות ב' 100000 פעמים והאות ג' 1000 פעמים.
ציירו את עץ הפמן של המחרוזת וחשבו את אורך הקידוד של המחרוזת לפי העץ הזה.

אורך הקידוד שווה ל: $10^6 + 2\cdot 10^5 + 2\cdot 10^3 = 1202000$



ב. שרטטו את ה-generalized suffix tree של המחרוזות **abba** ו-**bbca**. סמן בעץ את הצומת בו מסתיים החיפוש של המחרוזת bb.



ג. תארו אלגוריתם יעיל למציאת המחרוזת המשותפת הארוכה ביותר של שתי מחרוזות s1 ו-s2. מה סיבוכיות האלגוריתם שתיארתם?

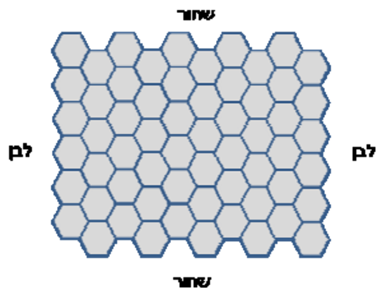
ראשית, בונים generalized suffix tree משתי המחרוזות.

עוברים על העץ בעזרת אלגוריתם רקורסיבי, מלמטה למעלה, ולכל קודקוד בודקים אם יש מתחתיו עלה ששייך ל-s1, וגם אם יש מתחתיו עלה ששייך ל-s2. בעזרת מעבר רקורסיבי גם מחשבים לכל קודקוד את ה-string-depth שלו (=אורך המחרוזת שמסתיימת בו).

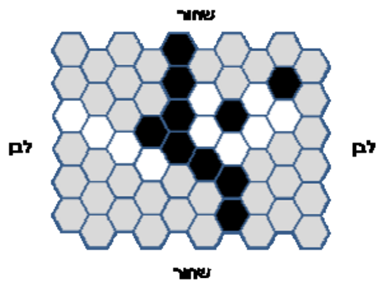
עכשיו עוברים על העץ, ומוצאים את הקודקוד v שיש לו string-depth מקסימלי, מבין כל אלה שיש להם צאצא שבא מ-s1 וגם צאצא שבא מ-s2. המחרוזת הכי ארוכה שמשותפת ל-s1 ול-s2 היא המחרוזת שמתקבלת כשמתחילים לקרוא בשורש, הולכים עד ל-v, וקוראים את המחרוזת שמורכבת ממה שכתוב על כל הקשתות שעברנו בדרך.

<u>שאלה 5 (8 נקודות)</u>

נגדיר משחק **הקס** שמשוחק על לוח של משושים. למשחק שני שחקנים (שחור ולבן), לכל אחד מהם משויכים שתיים מארבעת צלעות המשחק. כל אחד מהשחקנים מניח בתורו משושה (לבן או שחור) על משבצת פנויה בלוח. מנצח השחקן הראשון שמצליח לחבר את שני הצדדים שלו.



מצב התחלתי



השחור ניצח!

תכננו מבנה נתונים להחזקת הלוח כך שניתן יהיה <u>ביעילות</u> לעדכן את הלוח ולבדוק האם יש מנצח במהלך הנוכחי. על זמן העדכון והבדיקה לאחר כל מהלך לקחת $O(\log n)$ במקרה הגרוע ביותר (כאשר n הוא מספר המשושים על הלוח). פרט בקצרה באיזה פעולות תומך מבנה הנתונים וכיצד הוא ממומש.

נשמור שני מבני union-find. אחד לשחקן השחור, ואחד לשחקן הלבן. כל מבנה יאותחל לעבוד על יקום שבו רק שני אלמנטים: boundary-up ו-boundary-down עבור השחור, ו--boundary-left ו-boundary-right עבור הלבן. כל אחד יהיה בקבוצה בעצמו. בנוסף, מחזיקים עבור כל שחקן עץ אדום-שחור שמחזיק את התאים שהוא כבר תפס.

נתאר איך מעדכנים את מבנה הנתונים בהתאם למהלך של השחקן השחור. אם השחקן לוקח תא x, אז מכניסים את x לעץ של השחקן השחור, ומכניסים סינגלטון שמכיל רק את x למבנה ה-union-find. לאחר מכן בודקים איזה מן התאים השכנים ל-x נמצא בעץ, ולכל שכן y כזה, מבצעים פעולת union שלהקבוצה שמכילה את x עם הקבוצה שמכילה את y. בנוסף, אם x נמצא על השפה העליונה, מאחדים את הקב' שמכילה את x עם הקבוצה שמכילה את boundary-up. אם x נמצא על השפה התחתונה, מאחדים את הקב' שמכילה את x עם הקבוצה שמכילה את boundary-down. לבסוף, בודקים אם boundary-up ו-boundary-down נמצאים באותה קבוצה. אם כן, אז השחור ניצח.

את המבנים של השחקן הלבן מתחזקים בדומה.

זמן הריצה של כל פעולה יהיה $O(\log n)$, כי זו העלות גם של ההכנסה לעץ וגם של הפעולות על מבנה ה-union-find. זמן האתחול הוא $O(1)$.

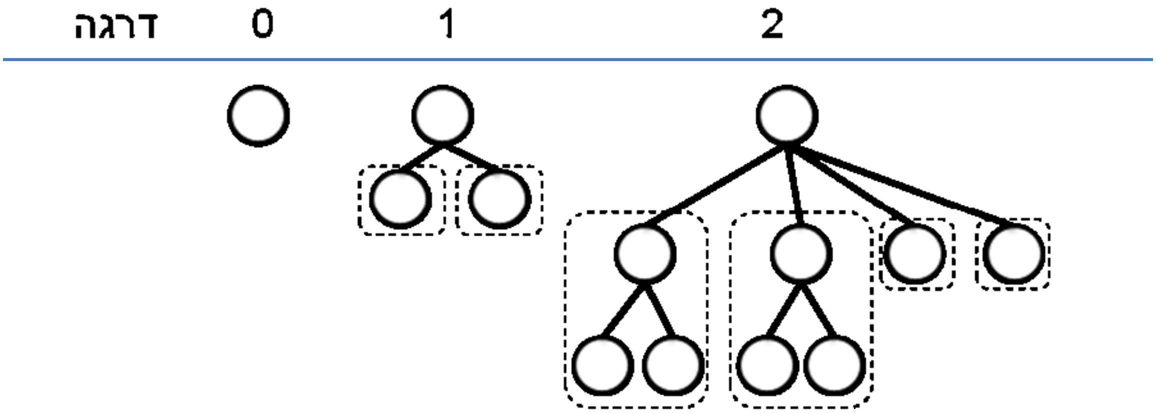
הערה: פתרון נכון נוסף הוא לשמור במקום העץ האדום-שחור פשוט טבלה של מצב כל הלוח. החסרון של הפתרון הזה הוא שהוא צורך $O(n^2)$ מקום וזמן איתחול.

שאלה 6 (13 נקודות) – שאלה 6 בגרסה 2

נגדיר עצים בינומיים "שמנים" בצורה הבאה:

עץ בינומי "שמן" מדרגה 0 מכיל צומת אחד בלבד.

עץ בינומי "שמן" מדרגה k ניתן לבנות משלושה עצים בינומיים "שמנים" מדרגה k-1 כאשר נתלה שניים מהם על העץ השלישי.



א. [6 נק'] תארו מבנה נתונים אנלוגי לערמה בינומית (binomial heap) המשתמש בעצים "שמנים".

ערמה בינומית "שמנה" מוגדרת בצורה הבאה: בערמה יש n עצים בינומיים "שמנים" כאשר יש לכל היותר 2 עצים מדרגה k, לכל k. שורשי העצים מחוברים ביניהם ברשימה מקושרת. הפעולות מוגדרות בדומה לערמה בינומית רגילה וכאשר יש צורך לעשות merge בין העצים (יש יותר משני עצים מדרגה k) בונים משלושה עצים מדרגה k עץ יחיד מדרגה k+1.

הערות בדיקה: רוב התלמידים כתבו נכון, מעט מאוד רשמו שמותר רק עץ אחד מכל דרגה

ב. [7 נק'] תארו פעולת meld של שתי ערמות בינומיות "שמנות" שהגדרתם בסעיף הקודם.

פעולת meld הינה מיזוג בין שתי ערמות בינומיות "שמנות". הדבר מקביל לחיבור שני מספרים בבסיס 3 (כמו שבערמות בינומיות הדבר היה מקביל לחיבור מס' בבסיס 2). נתחיל מדרגה 0 ונוסיף את העצים בערמה אחת לשנייה. אם יש סה"כ 3 או יותר עצים מדרגה 0 בערמה החדשה נמזג שלושה ליצירת עץ יחיד מדרגה 1. כעת נעבור לדרגה הבאה (1) ונמשיך כך...

הערות בדיקה: רוב התלמידים כתבו נכון.

שאלה 7 (10 נקודות)

ממפים n אלמנטים באמצעות משפחת פונקציות hash אוניברסאלית לטבלה עם $n^{5/3}$ תאים כאשר פותרים התנגשויות ע״י chaining.

א. מה תוחלת מספר הזוגות (x,y) כך ש-x ו-y הם אברים בקלט, ושניהם ממופים לאותו תא?

תשובה: $\Theta(\sqrt[3]{n})$.

נמספר את האיברים מ-1 עד n. יהי $X_{i,j}$ משתנה מקרי ששווה ל-1 אם איבר מס׳ i מתנגש עם איבר מס׳ j, ושווה ל-0 אחרת. יהי X הסכום של $\binom{n}{2}$ המשתנים הללו. אז X שווה למס׳ ההתנגשויות. לפי לינאריות התוחלת, $E[X]=\binom{n}{2}\cdot E[X_{1,2}]$. ההסתברות ששני איברים ספיציפיים מתנגשים היא 1 חלקי מס׳ התאים. לכן $E[X_{1,2}]=1/n^{5/3}$ ולכן

$$E[X]=\binom{n}{2}\Big/n^{5/3}=\frac{n-1}{2n^{2/3}}=\Theta(\sqrt[3]{n}).$$

ב. נניח כי הכנסנו את האיברים לטבלה, ומספר ההתנגשויות שקרו בפועל הוא לכל היותר פי 3 מהתוחלת שחישבת בסעיף א; במקרה זה, מה זמן החיפוש בטבלה במקרה הגרוע ביותר?

תשובה: $\Theta(\sqrt[6]{n})$.

הסבר: סיבוכיות w.c. של חיפוש שווה למס׳ האיברים בתא העמוס ביותר. נתון שיש לכל היותר $\Theta(\sqrt[3]{n})$ התנגשויות. שימו לב שאם בתא יש t איברים אז יש בו $\binom{t}{2}=\Theta(t^2)$ התנגשויות. המקרה הכי גרוע ב-w.c. הוא המקרה בו כל ההתנגשויות קורות בתא אחד. במקרה זה, בתא הנדון יש $\Theta(\sqrt[6]{n})$ איברים.

<u>שאלה 8 (10 נקודות) – שאלה 1 בגרסה 2</u>

נתונים n אלמנטים מתחום סדור במערך A. ידוע כי קיים אינדקס j כך שלכל i<=j מתקיים $A[i]<A[i+1]$ ולכל i>j מתקיים $A[i+1]<A[i]$.

דוגמה: j=6

1	2	4	5	6	9	26	15	14	3

א. תארו אלגוריתם יעיל למציאת j.

נתאר אלגוריתם הפועל בסיבוכיות $O(\log n)$. האלגוריתם מאוד דומה לחיפוש בינארי במערך ממוין. נסתכל על האבר ה-n/2. אם זהו האבר המתאים אזי סיימנו (גדול משני שכניו), אחרת אם הוא גדול משכנו הימני וקטן משכנו השמאלי, נחזור על הפעולה בחצי השמאלי של המערך רקורסיבית (על n/2 אברים). אחרת נחזור על הפעולה בחצי הימני של המערך רקורסיבית.

בכל שלב אנו מקטינים את מס' האברים הנבדקים פי 2, ולכן כפי שראינו פעמים רבות הסיבוכיות לוגריתמית.

הערות בדיקה: רוב התלמידים פתרו נכון, כמה חיפשו בכל שלב את החציון ולא את התא האמצעי. חציון ניתן למצוא רק בזמן לינארי ולכן זה לא מתאים!!

ב. הוכיחו כי $\Omega(\log n)$ הוא חסם תחתון על מספר ההשוואות הנדרש למציאת j.

אופציה א׳: נבנה עץ השוואות, סה״כ יש n אפשרויות עבור המיקום של j, ולכן בעץ צריכים להיות לפחות n עלים. בכל צומת בעץ ההשוואות אנו יכולים להשוות רק בין שני אברים ולכן אם העץ מאוזן עומקו הוא לוגריתמי

אופציה ב׳: רדוקציה לחיפוש בינארי. נניח כי ניתן למצוא את j בזמן o(logn) (ממש קטן מ-logn). אזי בהינתן מערך ממוין A ואבר x אותו אנו מחפשים נפעיל אלגוריתם למציאת j , כאשר נתייחס לכל אבר גדול מ-x כאל מספר שלילי. כך לדוגמה נתייחס למערך 1,3,5,7,9,15 כאשר x=7 כאל המערך 1,3,5,7,-9,-15. כעת נפעיל את האלג׳ ונקבל את המיקום של x. אך אנו יודעים שהחסם התחתון על חיפוש בינארי הינו לוגריתמי, וזאת בסתירה להנחה.

הערות בדיקה: היו כאן כמה טעויות חוזרות

- הוכיחו שהאלגוריתם בסעיף א׳ עובד בזמן לוגריתמי במקרה הגרוע ביותר (נוסחת רקורסיה). זה היה צריך להיכתב בסעיף א׳ ולא כאן.
- ניסו לעשות רדוקציה למיון של מערך. זה התחלק לשתי טעויות עיקריות
  - הניחו שA מערך כמו שמתואר והראו שניתן למיינו בזמן ממש קטן מ-(nlogn), אבל מערך כמו שמתואר ניתן למיין בקלות בזמן לינארי (למצוא את j ואז למזג את שני תתי המערכים)
  - הניחו שA מערך כללי, אבל אז לא ניתן להפעיל את הפונקציה, היא מסתמכת על כך שלA מבנה מאוד מסוים!

מספר תעודת זהות: [ ] 1 מתוך 1 מועד ב', גירסה 1

מספר מחברת: [ ]

מספר קורס: 0368.2158

סמסטר ב', מועד ב', תשס"ז

18.10.2007

ו' חשון, תשס"ח

מבחן במבני נתונים

פרופ' חנוך לוי, דני פלדמן ועודד שוורץ

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד A4 דו-צדדי.

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. הסברים לשאלות האמריקאיות – **חובה**!!!
4. תשובות סופיות לשאלות האמריקאיות יש לכתוב על הטופס המצורף.
5. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. **אין** להשתמש בפסאודו-קוד (אלא אם צוין אחרת). יש להסביר (באופן משכנע) את הרעיון הכללי.
8. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות.
9. גם אם שאלה נראית לכם דומה לשאלה מבחינות קודמות, היא לא בהכרח זהה.

בהצלחה!!

1. (7 נק') יהיו $f(n)=n^2, g(n)=(\lg\lg n)^{\lg\lg n}$.

נרצה להשוות את שתי הפונקציות. מה מהבאים נכון?

א. $f(n)=o(g(n))$

ב. $f(n)=\omega(g(n))$

ג. $f(n)=\Theta(g(n))$

ד. אף אחד מהנ"ל.

הסבר קצר (חובה) :

ב. בדיקה על ידי הוצאת lg לשתי הפונקציות.

2. (7 נק')

$$T(n)=\left(T(n-1)\right)^2 \qquad T(1)=2$$

$$S(n)=\sum_{i=1}^{\lg n}\lg\left(T(i)\right)$$

$S(n)=?$

א. $\Theta(\lg n)$

ב. $\Theta(\lg^2 n)$

ג. $\Theta(n)$

ד. $\Theta(n\lg n)$

ה. $\Theta(2^n)$

ו. אף אחד מהנ"ל.

הסבר קצר (חובה) :

ג.

$T(i)=2^{2^i}$ לכן $\lg T(i) = 2^i$

לכן S(n) הוא סדרה הנדסית שהאיבר הגדול ביותר שלה הוא n והיחס בין כל שנים עוקבים הוא 2.

מספר מחברת:

3. (7 נק') נתון מערך המכיל n נקודות במישור (כל נקודה נתונה על ידי זוג (x,y)). רוצים למצוא את $\frac{n}{\lg n}$ הנקודות הקרובות ביותר לישר $y = \frac{2007}{2006}x + 2007$. נרצה לעשות זאת בזמן ריצה W.C. מינימלי.

(רמז: ניתן לחשב מרחק של נקודה מקו בזמן O(1)).

זמן הריצה W.C. יהיה:

א. $\Theta\left(\frac{n}{\lg n}\right)$

ב. $\Theta(n)$

ג. $\Theta(n \lg n)$

ד. $\Theta(n \lg^2 n)$

ה. $\Theta(n^2)$

ו. אף אחד מהנ"ל.

הסבר קצר (חובה):

ב.

1. נחשב לכל נקודה את מרחקה מהישר. (זמן ריצה $\Theta(n)$).
2. נמצא על ידי Select את הסדר הסטטיסטי $n / \lg n$ של כל המרחקים. (זמן ריצה $\Theta(n)$).
3. נבדוק לכל מרחק האם הוא קטן מהסדר שנמצא ב-2. (זמן ריצה $\Theta(n)$).

4. (7 נק') נתון עץ אדום שחור עם $n > 2007$ מפתחות. העץ בנוי באופן חוקי, אבל לא בהכרח נוצר על ידי פעולות insert ו-delete. נסמן ב-R את מספר הצמתים האדומים וב-B את מספר הצמתים השחורים (לא כולל nil). מה מהבאים בהכרח נכון?

א. $B \le \left\lceil \frac{2006}{2007} n \right\rceil$

ב. $R \le \left\lceil \frac{2006}{2007} n \right\rceil$

ג. $R \le \left\lceil \frac{2}{3} n \right\rceil$

ד. ב + ג

ה. א + ב

ו. א + ב + ג.

הסבר קצר (חובה):

ד'. ייתכן עץ של מפתחות שחורים בלבד, לכן א' לא נכון.
לכל צומת אדום יש אבא שחור (למעט אולי השורש), ולאותו אבא שחור ייתכנו לכל היותר שני בנים אדומים. לכן על כל שני אדומים יש לפחות שחור אחד. כלומר $2R \le B + 1$. וכן $B + R = n$.
לכן $2R \le n - R + 1$ כלומר $R \le n/3 + 1$

5. (7 נק') נתון מערך A המכיל סריקת DLR של ערימת מקסימום בעלת n איברים. הערימה עצמה אינה נתונה. כיצד נבנה מערך B המכיל את הערימה בייצוג מערך?
(תזכורת: בייצוג מערך של ערימה הבנים, של האיבר במקום ה-i נמצאים במקומות 2i ו-2i+1).

א. פשוט נעתיק את A ל-B.

ב. א' לא נכון, אבל ניתן לבצע את השחזור בזמן $\Theta(n)$ W.C.

ג. ניתן לבצע את השחזור, אבל לא בזמן $\Theta(n)$ W.C.

ד. לא ניתן לבצע את השחזור, אבל היה ניתן לו קיבלנו גם את סריקת LDR של הערימה המקורית.

ה. אף אחד מהנ"ל.

הסבר קצר (חובה):
ב.
א' לא נכון. לא כל סריקת DLR של ערימה בגודל 4 עם 4 איברים שונים נותנת מערך שאינו ייצוג הערימה.
ב' נכון. ניקח מערך B ריק בגודל n. נתייחס אליו כאל ערימה, ונבצע עליו סריקת DLR. אבל במקום כל פעולת הדפסה של תא נעתיק אל התא את האיבר הבא מ-A.

6. (7 נק') נתון מבנה נתונים A המכיל n מפתחות. רוצים להדפיס את $\lceil \lg n \rceil$ האיברים הגדולים ביותר בזמן .W.C המהיר ביותר.

לשם כך עדיף ש-A יהיה:

א. טבלת Hash עם פתרון התנגשויות על ידי עצי 3-2.

ב. מערך שבו כל איבר נמצא במרחק לכל היותר $3\lceil \lg n \rceil$ ממקומו במערך הממוין.

ג. רשימה דו כיוונית ממוינת בסדר עולה.

ד. א + ג

ה. א + ב

ו. ב + ג

הסבר קצר (חובה): ו'.
א' – אפילו למצוא את הערך המקסימלי בלבד עלול לקחת $\Theta(n)$.
ב' – נסתכל על $3\lceil \lg n \rceil$ האיברים האחרונים במערך. נפריד מהם את $\lceil \lg n \rceil$ הגדולים ביותר בעזרת Select. סה"כ זמן ריצה $O(\lg n)$.
ג. ניגש לסוף הרשימה בעזרת המצביע Tail. נדפיס אחורה $\lceil \lg n \rceil$ איברים. זמן ריצה $O(\lg n)$.

7. (8 נק׳) מהו הגובה של עץ ההחלטות של אלגוריתם Select למציאת סדר סטטיסטי במערך בגודל n?

א. $\Theta(\lg n)$

ב. $\Theta(n)$

ג. $\Theta(n \log n)$

ד. $\Theta(n^2)$

ה. אלגוריתם Select לא מתאים למודל ההשוואות ולכן השאלה חסרת משמעות.

ו. אף אחד מהנ״ל.

הסבר קצר (חובה):

ב. ב–Select מבוצעות בדיקות על הקלט רק מסוג של השוואות ולכן מתאים למודל ההשוואות.

זמן ריצת Select הוא O(n) W.C. ולכן מספר ההשוואות W.C., כלומר גובה העץ, הוא O(n).

פעולת Partition הראשונה שמבוצעת ב–Select עלולה לדרוש Ω(n) השוואות, ולכן גובה העץ, הוא Ω(n).

8. (7 נק׳) נתונים n מפתחות, ורוצים לשמור אותם Perfect-Hash. מה צריך להיות גודל הטבלה B כך שבסיכוי לפחות ½, כבר בשלב הראשון של בניית הטבלה לא יהיו התנגשויות בכלל. (בחרו את הערך הקטן ביותר הנכון עבור B).

א. $B = \Theta(\lg n)$

ב. $B = \Theta(n)$

ג. $B = \Theta(n^2)$

ד. $B = \Theta(n^4)$

ה. $B = \Theta(2^n)$

ו. בכל טבלת Hash ייתכנו התנגשויות, לכן אף אחד מהנ״ל לא בהכרח נכון.

הסבר קצר (חובה):

ג.

הסיכוי להתנגשות בהכנסת האיבר ה–i לטבלה בגודל B הוא לכל היותר (i – 1)/B.

לכן סה״כ הסכוי להתנגשות חסום על ידי

1/B · (1 + 2 + 3 + … + (n – 1) כלומר חסום על ידי n^2 / B.

I. (18 נק') נתונה ערימת מינימום במימוש עץ-מצביעים.

נרצה להוסיף פעולת מחיקה מהערימה. הפעולה delete(x) מקבלת מצביע לאיבר בערימה, ומוחקת אותו מהערימה. בסוף הפעולה הערימה צריכה להיות חוקית, גם מבחינה מבנה וגם מבחינת יחסי אבא-בן.

א. (10 נק') הציעו אלגוריתם שמבצע את הנ"ל בזמן כולל W.C. $\Theta(\lg n)$

(6 נק') האלגוריתם:

ניקח את y העלה הימני ביותר בשכבה הנמוכה ביותר ונשים אותו במקום x.

נשווה אותו לאביו ולבניו במקומו החדש.

אביו קטן מבניו (כי זו הייתה ערימת מינימום חוקית לפני הפעולה).

לכן תיתכן הפרת חוקיות עם אביו או עם בניו אבל לא בו זמנית גם וגם.

אם ההפרה כלפי מטה – נבצע על y פעולת Heapify.

אם ההפרה כלפי מעלה – נחליף אותו עם אביו ונמשיך בהחלפות כלפי מעלה עד שתיפתר ההפרה (אולי רק בשורש).

(4 נק') ניתוח זמן ריצה: (לא לשכוח – להראות גם O וגם Ω)

זמן הריצה חסום בשני המקרים על ידי קובה הערימה ולכן הוא $O(\lg n)$.
כדי לראות שהוא גם $\Omega(\lg n)$ במקרה הגרוע ביותר, נשים לב ש\דוגמה שבה האיבר הנמחק x הוא שורש הערימה, והעלה y הוא קטן מכל אברי הערימה (ואז פעולת Heapify דורשת זמן ריצה $\Omega(\lg n)$.

ב. (8 נק') הוכיחו שאין אלגוריתם בעל זמן ריצה worst case טוב יותר, בהנחת מודל ההשוואות:

נניח בשלילה שיש אלגוריתם עם זמן ריצה W.C. טוב יותר בהנחת מודל ההשוואות, ונראה איך ניתן למיין בזמן טוב יותר מהחסם התחתון למודל זה, ולכן נסיק סתירה.
נסמן ב-$f(n)$ את זמן הריצה W.C. של פעולת המחיקה, על ערימה בגודל n.

1. נבנה ערימת מינימום (בזמן $O(n)$).
2. נדפיס את שורש הערימה (בזמן $O(1)$).
3. נמחק את שורש הערימה (בזמן $f(s)$ כאשר s גודל הערימה הנכחי).
4. אם הערימה לא ריקה נחזור לשלב 2.

סה"כ זמן ריצה חסום מלמעלה על ידי $O(n) + n\cdot f(n)$
לכן אם $f(n) = o(\lg n)$ נקבל זמן ריצה למיון במודל ההשוואות $o(n \lg n)$.
סתירה.

II. (10 נק') נתון עץ חיפוש בינרי (לאו דווקא מאוזן) שבו בכל צומת יש גם את השדה size(x), שמכיל את מספר הצמתים בתת העץ ששורשו בצומת x.

רוצים אלגוריתם שמקבל מצביע לשורש העץ וכן מספר k ומדפיס את כל המפתחות של הצמתים שהשדה size שלהם גדול מ-k.

א. (8 נק')הציעו אלגוריתם שרץ בזמן $\Theta(t)$ W.C. , כאשר t מספר המפתחות שיודפסו.

(6 נק') תיאור האלגוריתם :

נתחיל עם x מצביע לשורש.

```
PrintKeys(x,k)
If x = nil then return
If size(x) > k then
   Print key[x]
   PrintKeys(left[x])
   PrintKeys(right[x])
return
```

(2 נק') ניתוח זמן ריצה :

האלגוריתם הנ"ל מבצע למעשה סריקת DLR על תת העץ של כל הצמתים שאמורים להדפיס וכן על הבנים שלהם. יש t צמתים מיועדים להדפסה, ולכל היותר 2t בנים שלהם שנגיע אליהם ולא נדפיס אותם.

לכן זוהי סריקת DLR על עץ בגודל לכל היותר 3t, כלומר זמן ריצה O(t).

ב. (2 נק') הסבירו למה לא יתכן אלגוריתם יעיל יותר.

הסבר:
הדפסת t *איברים לוקחת לפחות* $\Omega(t)$ *(כי כל הדפסה לוקחת לפחות יחידת זמן אחת).*

III. (15 נק')

א. (8 נק')

קלט: n מספרים שונים זה מזה, כך שהעוקב של כל מספר גדול לפחות פי 2 ממנו. המספרים בתחום 1 עד 3^n.

מטרה: למיין את המספרים.

הציעו אלגוריתם יעיל ככל שתוכלו.

תיאור האלגוריתם:

נשתמש באלגוריתם BinSort, אבל התאים לא יהיו בגדלים קבועים, אלא בגדלים שהם סדרה הנדסית.

כלומר,

התא הראשון יהיה עבור המספרים שגדולים שווים 1 וקטנים מ-2.

התא השני יהיה עבור המספרים שגדולים שווים 2 וקטנים מ-4.

התא ה-i יהיה עבור המספרים שגדולים שווים 2^{i-1} וקטנים מ-2^i.

מספר התאים הוא $\lg(3^n) = n \lg(3)$

ניתוח זמן ריצה:

יש סה"כ $\Theta(n)$ תאים, בכל אחד לכל היותר איבר אחד.

לכן סה"כ זמן ריצה במקרה הגרוע ביותר הוא $\Theta(n)$.

ב. (7 נק')

נתון מערך המכיל n נקודות במישור (כל נקודה נתונה על ידי זוג (x,y)). נתון כי אין זוג נקודות עם אותו ערך x, ואין זוג נקודות עם אותו ערך y.

רוצים מבנה נתונים התומך בפעולה LowerPoint(x1,x2).

הפעולה הנ"ל מחזירה את הנקודה (x,y) בעלת y המינימלי מבין כל הנקודות שלהן יש x בתחום [x1,x2].

רוצים שזמן .W.C של הפעולה הנ"ל יהיה טוב ככל הניתן.

שימו לב שמבנה הנתונים **<u>לא צריך</u>** לתמוך בפעולות insert ו-delete.

תיאור המבנה:

נבנה עצים מאוזנים שיחזירו לכל x1 ו-x2 את הנקודה מהקלט שההיטל שלה על ציר x הכי קרוב (מימין ומשמאל). בנוסף, לכל זוג x1 ו-x2 נשמור את נקודת הקלט עם ה-y הכי נמוך, בתוך, למשל, Perfect hash.

טעות נפוצה: להניח ש-x1 ו-x2 הם קורדינטות x של נקודות מהקלט, ואז, למשל, לבנות את כל $O(n^2)$ התשובות מראש.

תיאור הפעולה:

בהינתן זוג x1 ו-x2 נמצא את הנקודות x1' ו-x2' מהקלט שההיטל שלהן על ציר x הכי קרוב מימין ומשמאל ל-x1 ו-x2. נמצא בטבלה את y עבור x1',x2'.

ניתוח זמן ריצת הפעולה:

חיפוש בעץ מאוזן: $O(\lg n)$ W.C.

חיפוש בטבלה: $O(1)$ W.C.

מספר קורס: 0368.2158

סמסטר א', מועד א', תשס"ז

222.200.7

ד' אדר, תשס"ז

מבחן במבני נתונים

פרופ' חנוך לוי, דני פלדמן ועודד שוורץ

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד A4 דו-צדדי.

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. הסברים לשאלות האמריקאיות – **חובה**!!!
4. תשובות סופיות לשאלות האמריקאיות יש לכתוב על הטופס המצורף.
5. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. **אין** להשתמש בפסאודו-קוד (אלא אם צוין אחרת). יש להסביר (באופן משכנע) את הרעיון הכללי.
8. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות.
9. גם אם שאלה נראית לכם דומה לשאלה מבחינות קודמות, היא לא בהכרח זהה.

בהצלחה!!

מספר מחברת:

1. (7 נק') יהיו $f(n)=\sqrt{n}^{\sqrt{n}}, g(n)=(\sqrt{n})!$.

נרצה להשוות **<u>את ה-log</u>** של שתי הפונקציות. מה מהבאים נכון?

א. $\lg(f(n))=o(\lg(g(n)))$

ב. $\lg(f(n))=\omega(\lg(g(n)))$

ג. $\lg(f(n))=\Theta(\lg(g(n)))$

ד. אף אחד מהנ"ל.

הסבר קצר (חובה): ג.

$$\lg(f(n))=\lg\left(\sqrt{n}^{\sqrt{n}}\right)=\sqrt{n}\lg\left(\sqrt{n}\right)=\frac{1}{2}\sqrt{n}\lg n$$

$$\lg(k!)=\Theta(k\lg k) \quad \Rightarrow \quad \lg\left(\left(\sqrt{n}\right)!\right)=\Theta\left(\sqrt{n}\lg\sqrt{n}\right)$$

$$\Rightarrow \lg(f(n))=\Theta(\lg(g(n)))$$

2. (7 נק') פתרו את נוסחת הנסיגה הבאה:

$$T(n)=\left(T\left(\frac{n}{2}\right)\right)^2\cdot 2^n \qquad T(1)=2$$

$T(n)=?$

א. $2^{\Theta(\lg n)}$

ב. $2^{\Theta(n)}$

ג. $2^{\Theta(n\lg n)}$

ד. $2^{\Theta(n^2)}$

ה. $\Theta\left(2^{n!}\right)$

ו. אף אחד מהנ"ל.

הסבר קצר (חובה): ג.
אפשר להפעיל lg על שני הצדדים, להציב S(n) = lg (T(n) , ולקבל נוסחת נסיגה כמו של MergeSort.
אפשר גם לפתח בשיטת האיטרציה, ואז המעריך מתנהג כמו הסכום המצטבר בפיתוח של MergeSort.
ואפשר בעוד שיטות.

3. (7 נק') נתונה ערימת מקסימום בגודל n. שינו את המפתחות של $\frac{n}{4}$ מהעלים בערימה, כך שעכשיו יתכן שהערימה כבר לא חוקית. נרצה להפוך את הערימה לחוקית בזמן ריצה W.C. מינימלי. הניחו שיש גישה לאברי הערימה (כלומר הערימה לא קופסה שחורה).

זמן הריצה W.C. יהיה:

א. $\Theta(\lg n)$

ב. $\Theta(\lg^2 n)$

ג. $\Theta(n)$

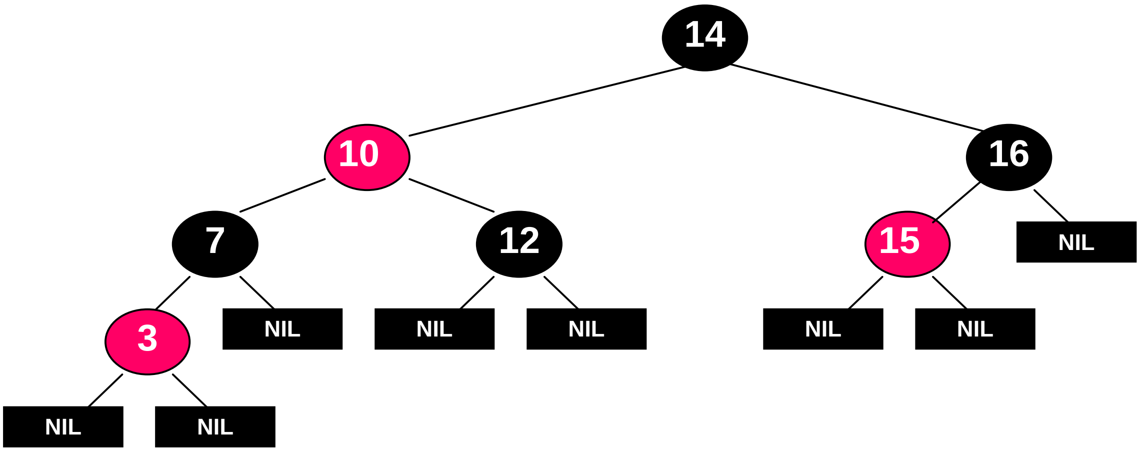
ד. $\Theta(n \lg n)$

ה. $\Theta(n^3)$

ו. אף אחד מהנ"ל.

הסבר קצר (חובה): ג.
הזמן שדרוש כדי לגשת לכל אחד מהצמתים ששושנו הוא $\Omega(n)$ לכן לא יתכן פתרון מהיר יותר.
בזמן $O(n)$ אפשר לבצע BuildHeap מחדש על כל האיברים.

4. (7 נקי) נתון עץ אדום שחור כמו שנלמד בכיתה (דוגמה מצורפת).



הניחו שהשורש שחור ושלכל צומת n מוסיפים שדה size(n) הסופר את מספר הצמתים בתת העץ ששורשו בצומת (לא כולל את צמתי ה-nil). הניחו שהעץ הוא אדום שחור חוקי (אך לא בהכרח נבנה בעזרת פעולות insert שנלמדו בכיתה). יהיו L,R בניו של השורש (הניחו שהם קיימים ואינם nil).

הננו מחפשים חסם ל size(R) כפונקציה של size(L). איזה חסם מהחסמים הבאים הינו החסם ההדוק ביותר על size(R) :

א. $\frac{1}{10}size(R) \le (size(L)+1)^2$

ב. $\frac{1}{10}size(R) \le 2(size(L)+1)$

ג. $\frac{1}{10}size(R) \le \sqrt{2}(size(L))$

ד. $\frac{1}{10}size(R) \le 2^{size(L)}$.

ה. $\frac{1}{10}size(R) \le 1.5(size(L)+1)$

הסבר קצר (חובה): א.
גובה השחור של L ו-R חייב להיות שווה בכל מסלול מעלה לשורש. לכן, העץ הקטן ביותר האפשרי לגובה שחור מסוים h יהיה עץ ללא צמתים אדומים (וגודלו יהיה 2^h).
מצד שני, לכל צומת אדום יש אבא שחור. לכן העץ המקסימלי מתקבל על ידי שכבות צמתים אדומות ושחורות לסרוגין. גודלו יהיה לא יותר מ- 2^{2h+1} (הגובה 2h+1 כי במקרה של L ו-R גם השורש יכול להיות אדום).

5. (7 נק') נבצע סריקת LDR על עץ חיפוש בינרי בגודל n (לא בהכרח מאוזן), אבל בכל שלב, במקום להדפיס את המפתח של הצומת הנכחי x, נדפיס את המפתח של אבא שלו. אם x הוא שורש העץ, לא נבצע הדפסה באותו השלב. מה מהבאים נכון?

א. כל מפתח בעץ, למעט המפתחות של העלים והשורש, יודפס פעמיים.

ב. הערכים שיודפסו, יהיו בסדר מונוטוני עולה.

ג. מספר המפתחות שיודפסו פעמיים הוא $\Theta(n)$ לכל עץ.

ד. א + ג

ה. א + ב + ג

ו. אף אחד מהנ"ל.

הסבר קצר (חובה): ו.
א' לא נכון כי כל צומת שיש לו בן יחיד יודפס רק פעם אחת.
ב' לא נכון, למשל העץ עם שורש 3, בן שמאלי 1 ולבן השמאלי בן ימני 2. יודפס קודם 3 ואחר-כך 1.
ג' לא נכון, למשל עבור עץ שבו כל צומת הוא עלה או שיש לו בן יחיד.

6. (7 נק') **<u>נתון</u>** מבנה נתונים A המכיל n מפתחות. רוצים למצוא **<u>מצביע</u>** לאיבר שהמפתח שלו הוא חציון המפתחות, בזמן .W.C המהיר ביותר. לשם כך עדיף ש-A יהיה:

א. טבלת Hash בגודל n עם פתרון התנגשויות על ידי עצי 2-3.

ב. רשימה דו כיוונית מקושרת ממוינת בסדר עולה.

ג. עץ AVL עם תמיכה בפעולות סדר-סטטיסטי.

ד. טבלת Hash בגודל n עם פתרון התנגשויות על ידי רשימות מקושרות דו כיווניות, וידוע שהערכים הם המספרים השלמים 1 עד n.

ה. א + ד

ו. ב + ג

הסבר קצר (חובה):ג.
א – צריך לסרוק את כל האיברים שבעצים שבטבלה ולכן יקח $\Theta(n)$.
ב- כדי להגיע לאיבר שבאמצע הרשימה צריך לרוץ על חציה, לכן יקח $\Theta(n)$.
ג – מציאת סדר סטטיסטי n/2 ניתנת לביצוע בזמן $\Theta(\lg n)$.
ד – החציון הוא המספר n/2. מציאת התא $\Theta(1)$. חיפוש ברשימה $\Theta(n)$. סה"כ $\Theta(n)$.

7. (7 נק') מהו החסם התחתון במודל ההשוואות של זמן ריצה .W.C של אלגוריתם מיון, אם ידוע ש- $\sqrt{n}$ האיברים הקטנים כבר במקום הנכון?

א. $\Theta(\lg n)$

ב. $\Theta(n\log n)$

ג. $\Theta(n)$

ד. $\Theta(n\sqrt{n})$

ה. $\Theta(n\sqrt{\lg n})$

ו. אף אחד מהנ"ל.

הסבר קצר (חובה): ב.

עדיין צריך למיין מערך בגודל $k=n-\sqrt{n}$. מיון מערך בגודל k בהנחת מודל ההשוואות לוקח $\Omega\left(k\lg k\right)$

אבל $\Omega\left(k\lg k\right)=\Omega\left(\left(n-\sqrt{n}\right)\lg\left(n-\sqrt{n}\right)\right)=\Omega\left(n\lg n\right)$.

8. (8 נק') נתון מערך המכיל יצוג של n נקודות במישור (כל נקודה נתונה על ידי זוג (x,y)). רוצים להכריע האם קיים ישר שעובר דרך ראשית הצירים (דרך (0,0)) כך שלפחות $\frac{3}{4}n$ מהנקודות נמצאות על הישר. נרצה אלגוריתם שפותר את הבעיה בזמן .W.C מינימלי. זמן .W.C יהיה:

א. $\Theta(\lg n)$

ב. $\Theta(n)$

ג. $\Theta(n\log n)$

ד. $\Theta(n^2)$

ה. $\Theta(n^2\lg n)$

ו. אף אחד מהנ"ל.

הסבר קצר (חובה): ב.

לכל נקודה נחשב בזמן $\Theta(1)$ את השיפוע של הישר המחבר אותה לראשית הצירים.

קיבלנו בזמן $\Theta(n)$ מערך המכיל את כל השיפועים, ונרצה לדעת האם יש ערך שחוזר $\frac{3}{4}n$ פעמים.

אם יש כזה הוא חייב להיות החציון. לכן, נפעיל את אלגוריתם Select למציאת החציון בזמן $\Theta(n)$,

ונוודא על ידי סריקת המערך שערך זה אכן מופיע $\frac{3}{4}n$ פעמים. אם כן – נחזיר "כן" אחרת נחזיר "לא".

סה"כ זמן ריצה $\Theta(n)$. ברור שאי אפשר מהר יותר כי צריך לקרוא את הקלט.

I. (18 נק') נתונה ערימת מינימום, **ללא מפתחות כפולים.**

לכל צומת x יש שדה נוסף s, שמאותחל ל-nil. רוצים שהשדה s של כל צומת x יכיל מצביע לעוקב של x. הניחו שיש גישה לאברי הערימה (כלומר הערימה לא קופסה שחורה). אין לשנות את מבנה הערימה.

א. (8) תנו חסם תחתון לזמן ריצה W.C. לפתרון הבעיה הנ"ל במודל ההשוואות.

הוכחה: נראה חסם $\Omega(n\lg n)$. נסמן ב-$T(n)$ את זמן הריצה W.C. של האלגוריתם הטוב ביותר לבעיה הנ"ל שפועל בהנחת מודל ההשוואות. נראה איך אפשר לבנות מאלגוריתם כזה אלגוריתם מיון:

ניקח מערך בגודל n. נבנה ממנו ערימה בזמן W.C. $\Theta(n)$.

נוסיף את המצביעים s בזמן $T(n)$.

נמצא את המינימום בזמן $\Theta(n)$.

נסרוק את הערימה בעזרת המצביעים - נתחיל מהדפסת המינימום ובכל צעד נעבור לאיבר הבא בעזרת המצביע s ונדפיס אותו.

קיבלנו אלגוריתם מיון שרץ בזמן $\Theta(n)+T(n)$.

כל הפעולות שביצענו על המפתחות המקוריים הן פעולות השוואה (גם בתוך האלגוריתם BuildHeap). לכן האלגוריתם מתאים להנחת מודל ההשוואות.

לפי החסם שהראינו בכיתה אלגוריתם כזה דורש זמן W.C. $\Omega(n\lg n)$.

לכן בהכרח $T(n)=\Omega(n\lg n)$

<u>טעויות נפוצות:</u>
- הוכחת חסם עליון במקום תחתון (כלומר, הצעת אלגוריתם ספציפי שפותר את הבעיה בזמן הנ"ל)
- התעלמות מזמן בניית הערימה (ההוכחה בכיתה התחילה ממערך של n איברים, ולא ערימה שקיימת בה חוקיות מסוימת)
- הוכחה שלא יכול להיות שעדכון כל איבר ספציפי ייקח o(logn), כלומר, קיים איבר שעבורו ייקח יותר זמן. מזה לא נובע שאי אפשר לעדכן את כל האיברים בזמן o(nlogn).

ב. (10) הציעו אלגוריתם שמבצע את הנ"ל בזמן כולל W.C. כמה שיותר טוב, אם נתון **שהמפתחות הם שלמים בתחום 1 עד 2n.**

תיאור האלגוריתם:

I. נסרוק את הערימה (למשל DLR) ונעתיק את המפתחות למערך עזר. נזכור לכל מפתח מצביע למקומו המקורי בערימה.

II. נמיין את המערך על ידי שימוש באלגוריתם מיון בזמן לינארי, למשל CountSort (או BinSort).

III. נסרוק את המערך משמאל לימין, ולכל איבר ניגש (בעזרת המצביע) לצומת המתאים לו בערימה ונעדכן את השדה s שיכיל את המצביע שבאיבר הבא במערך.

טעויות נפוצות:

- העתקת המפתחות למערך בלי לזכור מצביעים למקום בערימה
- אם מפרטים את מימוש ה-CountSort ע"י העתקה למערך בגודל 2n, יש לזכור שחלק מהתאים ריקים (יש רק n איברים), ולכן השדה s יכיל מצביע לאיבר הבא במערך **ששונה מ-nil.**
- מיותר להוכיח חסם תחתון או עליון שלא נתבקשתם

ניתוח זמן ריצה:

I. $\Theta(n)$

II. $\Theta(n)$

III. $\Theta(n)$

סה"כ $\Theta(n)$.

נשים לב שאין סתירה לסעיף הקודם כי האלגוריתם הנ"ל לא במודל ההשוואות.

II. . (10) נתון עץ 2-3 שבו בכל צומת מצוי שדה size(x) שמכיל את מספר הצמתים בתת העץ ששורשו בצומת x. מעוניינים להדפיס את k השדות size הגדולים ביותר במבנה. מובטח ש: $k < \lg n$.

שימו לב: יתכן מצב שבין k הצמתים שיודפסו, לא יהיה הבן הימני של השורש.

הציעו אלגוריתם עם זמן ריצה W.C. הטוב ביותר שתוכלו.

(8) תיאור האלגוריתם:

בדומה לאלגוריתם עבור הדפסת k האיברים הגדולים בערימת מקסימום.

נשים לב שהשדה Size של צומת תמיד גדול מהשדה Size של בניו. לכן השדה Size של צומת יודפס רק אחרי שיודפס השדה Size של אביו.

נחזיק ערימת עזר A (ערימת מקסימום) שתכיל את המועמדים להדפסה. בכל שלב נוציא מ-A את המקסימלי, נדפיס אותו, ובמקומו נכניס לערימה את השדה Size של שני בניו בעץ המקורי (אם יש לו רק בן אחד אז רק אותו ואם הוא עלה בעץ אז לא נכניס במקומו אף אחד). בנוסף נזכור, לכל ערך שהכנסנו ל-A, מצביע למיקומו בעץ.

נעצור אחרי שהדפסנו k ערכים, כאשר מתחילים עם הכנסת שורש העץ ל-A.

טעויות נפוצות:

- לשכוח שכל איבר בערימה חייב להכיל מצביע לאיבר המקורי בעץ, כדי שנוכל לשלוף את בניו בבוא היום.
- מציאת המפתחות הכי קטנים במקום ה-Size. (אגב, אפשר לעשות זאת ב-O(n) ולא O(nk)!
- התעלמות מכך שגודל הערימה הוא עד 3k, או לחוששים, O(K), אבל לא סתם k
- כמעט כולם התעלמו ממצב שבו לצומת שנמחק מהערימה אין בנים (עלה). יש להסביר שזה לא יקרה בגלל ש-k<lgn ולכן לא נגיע לכזה עומק.

(2) ניתוח זמן ריצה:

בכל שלב מוציאים מ- A איבר אחד ומכניסים לכל היותר שניים. לכן A תמיד בגודל לכל היותר k. לכן כל פעול על הערימה (הכנסה או הוצאה) לוקחת זמן $O(\lg k)$.

סה"כ זמן ריצה: $O(k\lg k)$.

כיוון ש- $k < \lg n$ זמן הריצה הוא $O(\lg n \lg\lg n)$.

בפרט הנ"ל טוב יותר, למשל מ-$\Theta(n)$, שאפשר לקבל על ידי מציאת הסדר הסטטיסטי ה-n-k מבין ערכי ה-Size (על ידי אלגוריתם Select) והדפסת כל ערכי הSize שגדולים ממנו או שווים לו.

III (15 נק')

בכיתה למדנו איך למצוא את העוקב של צומת x בעץ חיפוש בינרי T שיש לו n צמתים – הפעולה Successor(x). בשאלה הזו נניח שכל המפתחות שונים זה מזה. x הוא מצביע לאיבר ופעולת Successor(x) מחזירה מצביע לעוקב של x.

א. (3) תנו PseudoCode (ללא הוכחת נכונות) של פעולת Successor(x).

```
Successor(x)
        If right[x] ≠ nil then return min(right[x])
        While x = right[parent[x]] do
                x = parent[x]
        return parent[x]
```

ב. (4) מהו זמן ריצה W.C של פעולת העוקב, **<u>כפונקציה של n בלבד</u>**? הראו חסם Θ. כלומר חשבו חסם עליון O לזמן הריצה .W.C ותנו דוגמה שמצדיקה את החסם Ω. שימו לב שהעץ לא בהכרח מאוזן.

מספר מחברת:

חסם O :
זמן הריצה של האלגוריתם הוא O(h) במקרה הגרוע ביותר:
או שבכל שלב יורדים צעד אחד בעץ (במקרה של הפעלת Min)
או שבכל שלב עולים צעד אחד בעץ (במקרה של הפעלת לולאת While).
כיוון שגובה העץ הוא לכל היותר n, נסיק שזמן הריצה הוא O(n) W.C.

חסם Ω:
אם העץ הוא למשל שורש עם בן ימני שהוא עץ מנוון שמאלי, אז זמן הריצה יהיה הזמן שלוקח למצוא את המינימום של תת עץ ימין שזה $\Omega(n)$.

ג. (6) מריצים את הפעולה Successor(x) n פעמים – פעם אחת על כל צומת בעץ (העוקב של האיבר המקסימלי מוגדר להיות nil). נרצה לחשב את הזמן הממוצע לפעולת Successor בסדרה הנ"ל. נסמן ב-*t(x)* את זמן הריצה של פעולת Successor על הצומת x.

חשבו את: $\frac{\sum_{x\in T} t(x)}{n}$ (במונחי Θ) ותנו הסבר קצר.

עבור עץ מסוים T, זמן הריצה של כל פעולות ה-Successor לא תלוי בסדר שבו הפעלנו אותם.
לכן אפשר להניח שהקריאות מבוצעות לפי סדר עולה של המפתחות.
נשים לב שבמקרה כזה הפעולות המבוצעות הפועל הן בדיוק כמו בסריקת LDR על T (כי עבור כל צומת x תכונות העוקב ומבנה עץ החפוש תגרום לכך שנסרוק קודם את הצמתים בתת העץ השמאלי אח"כ את x ובסוף בתת העץ הימני) ולכן סה"כ זמן הריצה הוא $\Theta(n)$.

לכן: $\dfrac{\sum_{x\in T} t(x)}{n} = \dfrac{\Theta(n)}{n} = \Theta(1)$

ד. (1) ענו שוב על סעיף ב', אבל עכשיו בהינתן שהעץ הוא אדום שחור.

הראינו בסעיף ב' שזמן הריצה חסום מלמעלה על ידי גובה העץ. כיוון שגובה עץ אדום שחור הוא O(lg n) נסיק שזמן ריצת האלגוריתם במקרה הגרוע הוא O(lg n).

נראה גם חסם תחתון $\Omega(\lg n)$: העוקב של השורש בעץ אדום שחור הוא המינימום בתת עץ ימין. אם העץ מלא בכל שכבה, אז המינימום של תת עץ ימין הוא בעומק $\Theta(\lg n)$ ולכן מציאת העוקב תיקח $\Omega(\lg n)$.

ה. (1) ענו שוב על סעיף ג', אבל עכשיו בהינתן שהעץ הוא אדום שחור.

אין הבדל. בדיוק הטיעון בסעיף ג' מראה **<u>שלכל עץ</u>** גם אם הוא אדום-שחור: $\dfrac{\sum_{x\in T} t(x)}{n} = \Theta(1)$

מספר קורס: 0368.2158

סמסטר ב׳, מועד ב׳, תשס״ו

כ״ד תשרי, תשס״ז 16.10.06

מבחן במבני נתונים

פרופ׳ חיים קפלן, מר אלעד ורבין, מר דני פלדמן

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד A4 דו-צדדי.

הוראות כלליות:

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. הסברים לשאלות האמריקאיות – **חובה**!!!
4. תשובות סופיות לשאלות האמריקאיות ***יש לכתוב על טופס הבחינה***.
5. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.
8. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות.

בהצלחה!!

1. (9 נק') נתון עץ אדום-שחור ריק. מבצעים עליו n פעולות Insert, כאשר בפעולת Insert נתון רק מצביע לשורש העץ, והאלמנט החדש. (כלומר, לא נתון מצביע למקום ההכנסה, או כל מידע נוסף).
מה מהבאים נכון?

א. זמן הריצה הכולל יהיה $\Theta(n)$ ומספר הרוטציות בהכרח $O(n)$.

ב. זמן הריצה הכולל יהיה $\Theta(n\lg n)$ ומספר הרוטציות בהכרח $O(n)$.

ג. זמן הריצה הכולל יהיה $\Theta(n)$ ומספר הרוטציות יכול להיות $\Omega(n\lg n)$.

ד. זמן הריצה הכולל יהיה $\Theta(n\lg n)$ ומספר הרוטציות יכול להיות $\Omega(n\lg n)$.

הסבר **(חובה)**:

2. (9 נק') נתון עץ חיפוש אדום-שחור כאשר בכל צומת v נמצא שדה Size שמכיל את מס' הצמתים בתת-העץ ששורשו v (כולל v). מבצעים post-order (LRD) על העץ, אבל במקום להדפיס את המפתחות, מדפיסים את השדות Size
(תזכורת: סריקת post-order היא סריקה בה קודם מבצעים רקורסיבית סריקת post-order של תת-העץ השמאלי, אחרי כן של תת העץ הימני, ולבסוף מדפיסים את הערך בצומת עצמו).

השאלה: מה מהבאים נכון?

א. המספר הראשון שיודפס הוא הקטן ביותר.

ב. המספר האחרון שיודפס הוא הגדול ביותר.

ג. המספרים יודפסו בסדר מונוטוני עולה.

ד. סכום המספר הלפני-אחרון שיודפס עם המספר הלפני-לפני-אחרון שיודפס קטן באחד מהמספר האחרון שיודפס.

ה. א+ב

ו. א+ב+ג

הסבר **(חובה)**:

3. (9 נק') בשאלה זו נעסוק באלגוריתם מיון Quicksort. לכל אורך השאלה נניח שהקלט הוא מערך באורך n, ונתעניין רק במערכים בהם כל האיברים שונים. כזכור, לכל אלגוריתם מיון מתאים עץ בינארי המתאר את ההשוואות שהאלגוריתם מבצע על כל הקלטים האפשריים.

נתעניין באלגוריתם המיון QuickSort כאשר איבר הציר (Pivot) נבחר להיות הראשון בקטע המערך עליו האלגוריתם פועל (ולא נבחר על ידי הגרלה), ובעץ ההשוואות המתאים.

א. מהו העומק **<u>הקטן</u>** ביותר של עלה בעץ השוואות זה כפונקציה של n?

תשובה והסבר:

ב. מהו העומק **<u>הגדול</u>** ביותר של עלה בעץ השוואות זה כפונקציה של n?

תשובה והסבר:

4. (9 נק׳) מריצים את אלגוריתם QuickSort. בכל שלב איבר הציר נבחר להיות החציון. אבל, את החציון מוצאים על ידי פונקציה חדשה בשם NewSelect, שזמן הריצה שלה במקרה הגרוע על מערך בגודל n הוא $O\left(n^{3/2}\right)$. זמן הריצה של אלגוריתם QuickSort שהגדרנו במקרה הגרוע יהיה:

א. $O(n)$

ב. $O(n\lg n)$

ג. $O\left(n^2\right)$

ד. $O\left(n^{3/2}\right)$

ה. $O\left(n^{3/2}\lg n\right)$

ו. $O\left(n^3\right)$

הסבר **(חובה)**:

5. (9 נק׳) נתונים n מספרים. רוצים לבנות מהם מבנה נתונים. איזה מהמבנים הבאים זמן הבניה worst-case יהיה **<u>הארוך</u>** ביותר?

א. טבלת Hash שבה פתרון התנגשויות על ידי רשימות מקושרות.

ב. Perfect Hash

ג. ערימת מקסימום.

ד. עץ חיפוש AVL.

ה. א+ג

ו. א+ב+ג

הסבר **(חובה)**:

6. (9 נק׳) תאר מימוש לפעולה DeleteMin בערימה (Heap) הממומשת כמו שתיארנו בכתה שמבצע $O(\sqrt{\log n})$ השוואות במקרה הגרוע ביותר, או הוכח כי אין מימוש כזה.

תשובה:

7. א. (12 נק')

נתונים n קטעים על הישר (כל קטע נתון על ידי נקודות התחלה וסיום). לשם פשטות, נניח שכל 2n הנקודות שונות זו מזו.

קטע יקרא "מרכזי" אם יש לפחות $\frac{n}{4}$ קטעים שמתחילים לפניו (נקודת ההתחלה שלהם משמאל לנקודת ההתחלה שלו) וגם יש לפחות $\frac{n}{4}$ קטעים שמסתיימים אחריו (נקודת הסיום שלהם מימין לנקודת הסיום שלו).

הציעו אלגוריתם שמחזיר את **מספר** הקטעים המרכזיים, בזמן ריצה worst-case הטוב ביותר שתוכלו.

ב. (11 נק')

הציעו מבנה נתונים דינמי, המאפשר Insert ו-Delete של קטעים על הישר בזמן O(log n) במקרה הגרוע וכן תומך בפעולה IsCentric(p) שמקבלת מצביע לקטע במבנה, ועונה האם הקטע הוא מרכזי, בזמן O(1) במקרה הגרוע.

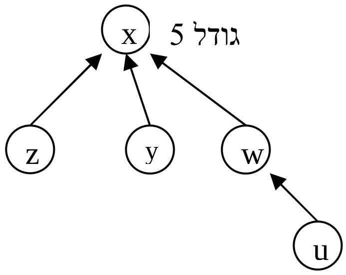
תאור המבנה:

מימוש פעולת Insert וניתוח זמן ריצה:

8. נממש מבנה נתונים המייצג אוסף של קבוצות זרות ותומך בפעולות הבאות:

Makeset(x): בהינתן איבר x בונה ומחזיר קבוצה המכילה רק את האיבר x.
Union(s1,s2): בהינתן שתי קבוצות s1 ו-s2 מחזיר קבוצה s שהיא איחוד של s1 ו-s2. הקבוצות s1 ו-s2 אינן קיימות לאחר ביצוע הפעולה.
Find(x): בהינתן מצביע לצומת המייצג איבר z, מחזיר את הקבוצה בה נמצא z. (שים לב – מניחים שמתקבל מצביע ישירות **לצומת** המייצג את z)

מייצגים כל קבוצה על-ידי עץ. בכל צומת בעץ נמצא איבר ונמצא מצביע לאביו בעץ. בשורש העץ מאוכסן גם גודל הקבוצה. שורש העץ משמש כ"מייצג" של הקבוצה. לדוגמה: העץ



מייצג את הקבוצה {x,y,z,w,u} שגודלה 5.

מימוש Makeset מקצה צומת חדש שמהווה עץ חדש. בשורש נאחסן את האיבר x, ואת הגודל שהוא 1.
מימוש Union(s1,s2) יתבצע על-ידי הפיכת השורש של הקבוצה הקטנה יותר לבן של השורש של הקבוצה הגדולה יותר. כמו כן נשנה את הגודל של השורש בקבוצה הגדולה יותר להיות סכום גדלי שתי הקבוצות.
מימוש find(z) יתבצע על-ידי כך שנתקדם מהצומת z על-פי מצביעי האב עד לשורש ונחזיר את השורש.

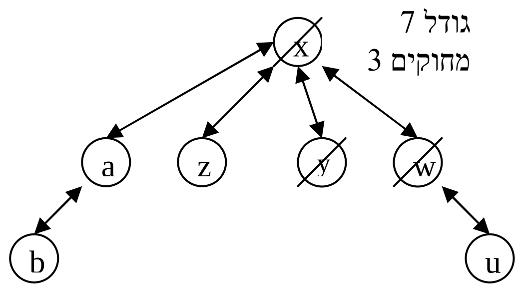
א. (12 נק') מהו החסם על סיבוכיות הזמן במקרה הגרוע של כל פעולה. הוכח תשובתך.

תשובה:

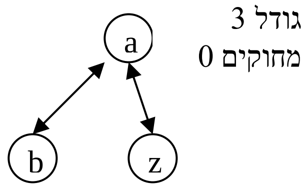
ב. (11 נק') מוסיפים גם את הפעולה Delete(z) שמוחקת את האיבר המאוחסן בצומת אליו מצביע z. (שוב – כמו ב-Find, מקבלים מצביע **ישירות** ל-z). כדי לתמוך בפעולה זו נרשה צמתים שבהם נמצאים איברים המסומנים כמחוקים (deleted). בשורש של קבוצה נרשום גם את מספר האיברים המחוקים בעץ. כמו-כן, בסעיף זה כל צומת יחזיק רשימת מצביעים לבניו בנוסף למצביע לאביו. (במימוש שתואר בסעיף הקודם לא היו מצביעים לבנים, רק לאב).

המימוש של Delete(z) מסמן את האיבר בצומת z כמחוק, ואחר-כך עוקבים אחר מצביעי-האב עד לשורש ומגדילים את מספר האיברים המחוקים בקבוצה ב-1. אם מספר האיברים המחוקים גדול מחצי מספר אברי הקבוצה, סורקים את כל העץ המייצג את הקבוצה, ומשחררים כל צומת המחזיק איבר מחוק. מבין האיברים שאינם מחוקים, בוחרים אחד שיהיה השורש, ואת כל שאר האיברים הלא-מחוקים הופכים להיות בנים שלו.

לדוגמה, אם הקבוצה s={z,u,a,b} מיוצגת ע"י המבנה בצורה הבאה:



ומבצעים Delete(u) אזי העץ שמייצג את הקבוצה לאחר ביצוע הפעולה יהיה:



בהנחה ש-a נבחר להיות השורש.

כמה זמן לוקח לבצע סדרה של m פעולות, כאשר מספר האלמנטים בכל רגע לא עולה על n, וכאשר מס' הפעולות שהן Find או Delete הוא k? (שאר m-k הפעולות הן Makeset או Union). הוכח את תשובתך.

מספר תעודת זהות:

מספר מחברת:

מועד ב', גירסה 1

מספר קורס: 0368.2158

סמסטר א', מועד ב', תשס"ו

כ"ד תשרי, תשס"ז 16.10.06

מבחן במבני נתונים

פרופ' חנוך לוי, עודד שוורץ

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד A4 דו-צדדי.

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. הסברים לשאלות האמריקאיות – **חובה**!!!
4. תשובות סופיות לשאלות האמריקאיות יש לכתוב על הטופס המצורף.
5. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.
8. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות.

בהצלחה!!

מספר תעודת זהות: ☐ 3 מתוך 10 מועד ב', גירסה 1

מספר מחברת: ☐

1. (7 נק') יהיו $f(n)=(n!)^{2006}, g(n)=(n^{2006})!$.

נרצה להשוות **<u>את ה-log</u>** של שתי הפונקציות. מה מהבאים נכון?

א. $\lg(f(n))=o(\lg(g(n)))$

ב. $\lg(f(n))=\omega(\lg(g(n)))$

ג. $\lg(f(n))=\Theta(\lg(g(n)))$

ד. אף אחד מהנ"ל.

הסבר קצר:

2. (7 נק') פתרו את נוסחת הנסיגה הבאה:

$$T(n)=3T(n-2)+9$$
$$T(1)=T(2)=1$$

$$T(n)=?$$

א. $\Theta(n^3)$

ב. $\Theta(n^{\log_3 9})$

ג. $\Theta(3^n)$

ד. $\Theta(n^{1.5})$

ה. $\Theta((\sqrt{3})^n)$

הסבר קצר:

3. (8 נק')

נתון עץ AVL ריק. מבצעים עליו n פעולות Insert. מה מהבאים נכון?

א. זמן הריצה הכולל יהיה $\Theta(n)$ ומספר הרוטציות בהכרח $O(n)$.

ב. זמן הריצה הכולל יהיה $\Theta(n\lg n)$ ומספר הרוטציות בהכרח $O(n)$.

ג. זמן הריצה הכולל יהיה $\Theta(n)$ ומספר הרוטציות יכול להיות $\Omega(n\lg n)$.

ד. זמן הריצה הכולל יהיה $\Theta(n\lg n)$ ומספר הרוטציות יכול להיות $\Omega(n\lg n)$.

הסבר קצר :

4. (8 נק') נתון עץ חיפוש AVL התומך בפעולות סדר סטטיסטי (Order Statistic), כלומר בכל צומת יש גם את השדה Size. מבצעים סריקת LRD על העץ, אבל במקום להדפיס את המפתחות, מדפיסים את השדות Size. מה מהבאים נכון?

א. המספר הראשון שיודפס הוא הקטן ביותר.

ב. המספר האחרון שיודפס הוא הגדול ביותר.

ג. המספרים יודפסו בסדר מונוטוני עולה.

ד. א+ב

ה. א+ב+ג

הסבר קצר :

5. (8 נק') נתעניין באלגוריתם המיון QuickSort כאשר איבר הציר (Pivot) נבחר להיות הראשון בקטע המערך עליו האלגוריתם פועל (ולא נבחר על ידי הגרלה).

מהו העומק **<u>הקטן</u>** ביותר של עלה בעץ ההחלטות של אלגוריתם זה על מערך בגודל n?

א. $\Theta(n)$

ב. $\Theta(n\log n)$

ג. $\Theta(n^2)$

ד. $\Theta(n!)$

ה. $\Theta(n!\lg n)$

ו. האלגוריתם הנ"ל לא מתאים למודל ההשוואות ולכן השאלה חסרת משמעות.

הסבר קצר:

6. (8 נק') רוצים להריץ את אלגוריתם QuickSort עם השינוי הבא: איבר הציר יבחר על ידי מציאת החציון, בעזרת פונקצית NewSelect. זמן הריצה של NewSelect על מערך בגודל n הוא W.C. $\Theta\left(n^{\frac{2006}{2005}}\right)$. זמן הריצה של אלגוריתם QuickSort החדש יהיה:

א. W.C. $\Theta(n)$

ב. W.C. $\Theta(n\lg n)$

ג. W.C. $\Theta(n^2)$

ד. W.C. $\Theta\left(n^{\frac{2006}{2005}}\right)$

ה. W.C. $\Theta\left(n^{\frac{2006}{2005}}\lg n\right)$

ו. W.C. $\Theta\left(n^{2\cdot\frac{2006}{2005}}\right)$

הסבר קצר:

7. (8 נק') נתונים n איברים. רוצים לבנות מהם מבנה נתונים. איזה מהמבנים הבאים זמן הבניה W.C. יהיה **<u>הארוך</u>** ביותר?

א. טבלת Hash שבה פתרון התנגשויות על ידי רשימות מקושרות.

ב. Perfect Hash

ג. ערימת מקסימום.

ד. עץ חיפוש AVL.

ה. א+ג

ו. א+ב+ג

הסבר קצר:

מספר תעודת זהות:

מספר מחברת:

שאלות פתוחות:

I. (15 נק׳)

נתונים n קטעים על הישר (כלומר הקלט הוא n זוגות (x,y) של נקודות התחלה וסיום).

קטע יקרא ״מרכזי״ אם יש לפחות $\frac{n}{4}$ קטעים שמתחילים לפניו (נקודת ההתחלה שלהם משמאל לנקודת ההתחלה שלו) וגם יש לפחות $\frac{n}{4}$ קטעים שמסתיימים אחריו (נקודת הסיום שלהם מימין לנקודת הסיום שלו).

הציעו אלגוריתם שמחזיר את **מספר** הקטעים המרכזיים, בזמן ריצה .W.C הטוב ביותר שתוכלו.

אלגוריתם:

ניתוח זמן ריצה:

II. (15 נק׳)

הציעו מבנה נתונים דינמי, המאפשר Insert Find ו-Delete של קטעים על הישר בזמן O(lg n) W.C. וכן בפעולה IsCentric(p) שמקבלת מצביע לקטע במבנה, ועונה האם הקטע הוא מרכזי, בזמן O(1) W.C.

תאור המבנה:

מימוש פעולת Insert וניתוח זמן ריצה:

מימוש פעולת Delete וניתוח זמן ריצה:

מימוש פעולת IsCentric וניתוח זמן ריצה:

מספר תעודת זהות:

מספר מחברת:

10 מתוך 10

מועד ב', גירסה 1

III. (15 נק') הוכיחו שבהנחת מודל ההשוואות, לא קיים מימוש משופר לפעולת ExtractMax (הוצאת האיבר המקסימלי והבאת הערימה למצב ערימה חוקית) בערימת מקסימום, שמבטיח זמן ריצה $o(\lg n)$ W.C. (שימו לב – ה-o קטן).

הוכחה:

מספר קורס: 0368.2158

סמסטר א', מועד א', תשס"ו

ב' אדר, תשס"ו, 2/3/2006

מבחן במבני נתונים – הצעת פתרון

פרופ' חנוך לוי, עודד שוורץ

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

הוראות כלליות:

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף.
2. המחברות הן לטיוטה בלבד ולא תיבדקנה.
3. הסברים לשאלות האמריקאיות – **חובה!!!**.
4. תשובות סופיות לשאלות האמריקאיות יש לכתוב על הטופס המצורף.
5. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.
8. כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות, ארוכות, מסובכות או מסורבלות יקבלו ניקוד חלקי גם אם הן נכונות.

בהצלחה!!

	גירסה 1	גירסה 2		גירסה		גירסה 4	
שאלה	פתרון	מתאימה לשאלה בגירסה 1	פתרון	מתאימה לשאלה בגירסה 1	פתרון	מתאימה לשאלה בגירסה 1	פתרון
1	א	2	ג	2	ג	2	ג
2	ג	1	א	1	א	8	ה
3	ה	4	ה	6	ג	4	ה
4	ה	3	ה	3	ה	3	ה
5	ו	5	ו	4	ה	5	ו
6	ג	6	ג	7	א	7	א
7	א	7	א	8	ה	6	ג
8	ה	8	ה	5	ו	1	א

1. **(8 נקודות)** יהיו $f(n)=(n!)^{2006}, g(n)=(n^{2006})!$.

נרצה להשוות את שתי הפונקציות. מה מהבאים נכון?

א. $f(n)=o(g(n))$

ב. $f(n)=\omega(g(n))$

ג. $f(n)=\Theta(g(n))$

ד. ב + ג נכונים

ה. א + ג נכונים

ו. אף אחד מהנ"ל אינו נכון.

2. **(8 נקודות)** נתונה נוסחת הנסיגה הבאה: $T(n)=T\left(\frac{n}{3}\right)+T(\alpha\cdot n)+n$

מבין הערכים הבאים עבור α, מהו הגבוה ביותר שעבורו מתקיים ש- $T(n)=O(n)$?

א. $\alpha=\frac{1}{4}$

ב. $\alpha=\frac{1}{3}$

ג. $\alpha=\frac{1}{2}$

הסבר קצר(**חובה**): א.
f(n) היא מכפלה של *2006n* גורמים שכולם קטנים או שווים *n*.
g(n) היא מכפלה של n^{2006} גורמים, שכולם למעט *n* מהם גדולים מ-*n*.

Formatted

ד. $\alpha=\frac{2}{3}$

ה. $\alpha=\frac{3}{4}$

ו. $\alpha=\frac{2005}{2006}$

Formatted

הסבר קצר (**חובה**): ג.
אם α הוא קבוע קטן מ-*2/3* אז $T(n)=O(n)$ (כמו שראינו בש"ב).
אם α הוא קבוע *2/3* אז $T(n)=\Theta(n\lg n)$ (ניתוח בעזרת למשל עץ רקורסיה בדומה לחישוב זמן הריצה של *MergeSort*).

3. (**8 נקודות**) נוסחת הנסיגה $T(n)=T\left(\left\lceil\frac{n}{2}\right\rceil\right)+T\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+\Theta(n)$

מתארת את זמן W.C. הריצה של האלגוריתמים הבאים:

הערה: עבור אלגוריתם מסוים **לא מספיק שסיבוכיות** תהיה זהה לפתרון נוסחת הנסיגה. צריך גם שנוסחת הנסיגה תתאים ל**פעולת האלגוריתם.**

א. QuickSort

ב. QuickSort כאשר איבר הציר (Pivot) נבחר להיות החציון על ידי Select.

ג. מיון על ידי הכנסות לעץ חיפוש AVL וסריקת העץ בסוף.

ד. MergeSort

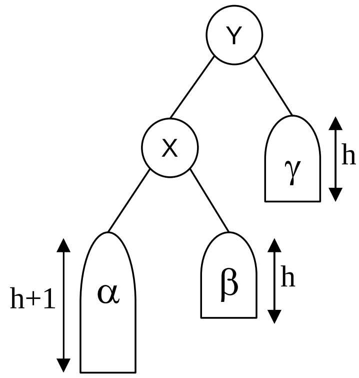
ה. ב + ד

ו. ב + ג + ד

Formatted: Underline

הסבר קצר(**חובה**): ה.
נוסחת הנסיגה מתארת את *MergeSort* וגם את *QuickSort* בתנאי שאיבר הציר נבחר להיות החציון בזמן ליניארי (למשל על ידי *Select*). ב-*QuickSort* רגיל החלוקה עלולה להיות לא מאוזנת ולכן נוסחת הנסיגה הנ"ל לא מתאימה. מיון על ידי עץ AVL אמנם לוקח $\Theta(n\lg n)$, שזהו הפתרון של T, אבל T לא מתאימה לתיאור זמן הריצה (למשל, כי זה לא אלגוריתם רקורסיבי).

4. **(8 נקודות)** נתון עץ חיפוש AVL הבא, רגע אחרי פעולת Insert אבל (כמו שרואים) לפני פעולת תיקון העץ:



נניח שבעץ הנ"ל בכל צומת יש גם את השדה Size (המבטא את גודל תת העץ ששורשו בצומת).
נסמן את הגודל של העצים α, β, γ ב - S_α, S_β, S_γ בהתאמה, וכן ב-S_x ו-S_y את הגודל של העצים שהשורש שלהם הוא X ו-Y בהתאמה (כל הגדלים הנ"ל אחרי פעולת ה-Insert ולפני תיקון העץ).
מה יהיה השדה Size[Y] אחרי פעולת ה-Insert ופעולת התיקון העץ?

א. $S_y + 1$

ב. $S_y + 2$

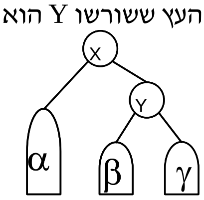
ג. $S_\alpha + S_\beta + 1$

ד. $S_\alpha + S_\beta + 2$

ה. $S_\beta + S_\gamma + 1$

ו. $S_\beta + S_\gamma + 2$

הסבר קצר(**חובה**): ה. התיקון הוא רוטציה. העץ אחרי הרוטציה יראה כמו בציור למטה. לכן הגודל של תת העץ ששורשו Y הוא הגודל של β ועוד הגודל של α ועוד אחד (בשביל השורש).



5. **(8 נקודות)** נתון עץ חיפוש 2-3 שבו לשורש יש בדיוק שני בנים: שמאלי ומרכזי.
נסמן ב-X את **מספר הצמתים** בתת העץ המרכזי וב-Y את **מספר הצמתים** בתת עץ שמאל. מה מהבאים בהכרח נכון?
(אם יש יותר מתשובה אחת נכונה, בחרו את ההדוקה ביותר).

א. $X = Y$
ב. $X \le \frac{3}{2}Y$
ג. $X \le 2Y$
ד. $X \le 6Y$
ה. $X \le 2006 \cdot Y$
ו. אף אחד מהנ"ל.

הסבר קצר(**חובה**): ו.
יתכן מצב שבו בתת עץ אחד לכל צומת יש בדיוק שני בנים ובתת העץ השני לכל צומת יש בדיוק 3 בנים. לכן אם גבה העץ הוא h אז בתת עץ אחד מספר הצמתים הוא $\sum_{i=0}^{h} 2^i$ ובשני $\sum_{i=0}^{h} 3^i$. כלומר בראשון יש 2^{h+1} צמתים ובשני יותר מ- 3^h צמתים.

6. **(8 נקודות)** נתייחס לאלגוריתם המיון הבא:
בדוק האם הקלט ממוין על ידי השוואת כל איבר לבא אחריו במערך.
אם כן – סיים. אחרת – הפעל את אלגוריתם MergeSort.

מהו העומק **המְמוצע** של עלה בעץ ההחלטות של אלגוריתם המיון הנ"ל על מערך בגודל n?

א. $\Theta(\lg n)$
ב. $\Theta(n)$
ג. $\Theta(n \log n)$
ד. $\Theta(n^2)$
ה. $\Theta(n!)$
ו. האלגוריתם הנ"ל לא מתאים למודל ההשוואות ולכן השאלה חסרת משמעות.

Formatted: Left-to-right

הסבר קצר(**חובה**): ג.
עבור קלט ממוין מספר ההשוואות שמבצעים הוא n. כלומר העלה שמייצג מערך ממוין הוא בעומק n.
לכל קלט אחר מספר ההשוואות ש-*MergeSort* מבצע הוא $\Theta(n\log n)$. כלומר כל עלה אחר הוא בעומק $\Theta(n\log n)$. יש בעץ יש $n!$ סה"כ. לכן העומק הממוצע הוא: $\frac{(n!-1)\cdot\Theta(n\log n)+1\cdot n}{n!} = \Theta(n\log n)$

7. **(8 נקודות)** נתון עץ חיפוש בינרי (לאו דוקא מאוזן) בעל n צמתים שבו לכל צומת זוג שדות נוספים: Pred, Suc שמצביעים לקודם ולעוקב של הצומת. רוצים לממש פעולת Delete על עץ זה בזמן .W.C הטוב ביותר האפשרי (כך שאחרי המחיקה כל השדות בעץ נכונים). פעולת המחיקה מקבלת מצביע לאיבר המיועד למחיקה. מותר להניח שקיימים כל השדות הרגילים, כמו Parent[x].
הזמן המתקבל הוא:

א. $\Theta(1)$

ב. $\Theta(\lg n)$ ללא תלות בגובה העץ.

ג. $\Theta(n)$ ללא תלות בגובה העץ.

ד. $\Theta(h)$ כאשר h גובה העץ.

ה. $\Theta(n \lg n)$

ו. לא ניתן לבצע מחיקה בעץ כזה, כי אי אפשר לקבוע בוודאות את הערך החדש לשדות Pred ו-Suc.

הסבר קצר(**חובה**): א.
נניח ש-x הוא האיבר הנמחק. נזכור את הקודם והעוקב של x כדי לעדכן את ההצבעה ההדדית שלהם אחרי מחיקת x.
אם ל-x יש בן יחיד או הוא עלה – נמחק אותו בזמן $\Theta(1)$ כמו במחיקה רגילה. אם ל-x יש שני בנים – נמצא את העוקב שלו בעזרת השדה *Suc* בזמן $\Theta(1)$, ונמשיך כמו במחיקה רגילה: החלפה ומחיקה במקום החדש. סה"כ זמן - $\Theta(1)$.

8. **(8 נקודות)** לאיזה מהמבנים הבאים יש זמן חיפוש .W.C **הגרוע** ביותר?

א. טבלת Hash שבה פתרון התנגשויות על ידי רשימות מקושרות.

ב. Perfect Hash

ג. ערימת מקסימום במימוש מערך.

ד. א+ב

ה. א+ג

ו. א+ב+ג

הסבר קצר(**חובה**): ה.
לטבלת *Hash* ולערימה זמן חיפוש *W.C.* $\Theta(n)$. ל-*Perfect Hash* זמן חיפוש $\Theta(1)$ *W.C.*.

שאלות פתוחות:

I. (14 נקודות)

נתונה ערימת מקסימום במימוש מערך (יש גישה לאיברי המבנה) בגודל n ונתון מספר k.

רוצים להדפיס את k האיברים הגדולים בערימה בזמן $\Theta(k\lg k)$ W.C.

רמז: אפשר להשתמש בערימה נוספת.

אלגוריתם והסבר נכונות:

האלגוריתם:
נחזיק ערימת עזר. בערימת העזר נחזיק חלק מהמפתחות של הערימה המקורית, עם מצביעים לצמתים המתאימים בערימה המקורית. בהתחלה נעתיק לערימת העזר את שורש הערימה המקורית.
בכל שלב נבצע *ExtractMax* מערימת העזר. נדפיס את האיבר שיצא x. את שני בניו בערימה המקורית נכניס לערימת העזר. נחזור על הנ"ל עד שהודפסו k איברים.
הסבר נכונות:
אם המפתח של צומת הוא לא בין k המפתחות הגדולים, אז לא צריך להדפיס אותו, ובהכרח גם את בניו לא צריך להדפיס. נשים לב שבכל שלב ערימת העזר מכילה את כל האיברים הפוטנציאלים להדפסה, כלומר את כל הבנים (מהערימה המקורית) של הצמתים שהודפסו. כל צומת שאביו יודפס – יכנס בעצמו לערימה, וכל צומת שאביו לא יודפס – לא יכנס לערימה. לכן האלגוריתם ידפיס את k הערכים הגדולים ביותר, ואפילו בסדר יורד.

ניתוח זמן ריצה:
בכל שלב מוציאים מערימת העזר איבר אחד ומכניסים במקומו שניים. כלומר גודל ערימת העזר הוא לכל היותר k+1. לכן זמן פעולה בסיסית על הערימה הוא $O(\lg k)$.
מציאת שני הבנים מבוצעת בזמן $O(1)$ (כי יש מצביע לאיבר המתאים בערימה המקורית). סה"כ $O(k)$.
בכל שלב של הריצה מבצעים שלוש פעולות על ערימת העזר (פעם אחת Extract ופעמיים Insert). יש k שלבים כאלה. סה"כ $O(k\lg k)$.
לכן סה"כ זמן ריצה $O(k\lg k)$.
נשים לב שב-k/2 השלבים האחרונים הערימה כבר בגודל לפחות k/2, לכן זמן פעולה בסיסית על הערימה הוא $\Omega(\lg k)$, כלומר זמן הריצה הכולל הוא גם $\Omega(k\lg k)$, ולכן $\Theta(k\lg k)$.

II. (22 נקודות)

א. (13 נקודות) תור-פרוטקציה הוא מבנה נתונים שתומך בפעולות של תור (Enqueue,Dequeue)
ובנוסף – פעולת כניסה בפרוטקציה PEnqueue.

פעולות Enqueue,Dequeue שומרות על סדר תור רגיל (הראשון שנכנס הוא הראשון שיוצא).
איבר שנוסף לתור בפעולת PEnqueue יצא מהתור לפני כל האיברים בתורשנכנסו בפעולת
Enqueue וגם לפני כל האיברים בתור שנכנסו (לפניו) בפעולת PEnqueue.

לדוגמה, הפלט בעקבות הפעולות הבאות:

Enqueue(3), Enqueue(4), Enqueue(10), Dequeue, Penqueue(2006), Penqueue(2005), Dequeue, Dequeue, Dequeue

יהיה (משמאל לימין)

3, 2005, 2006, 4

הציעו מימוש למבנה זה המשתמש **<u>במספר מינימלי של מחסניות</u>** בלבד וכן $O(1)$ זכרון נוסף.
על המחסנית מותר לבצע את הפעולות הרגילות אבל אין גישה למימוש.
זמן W.C. של פעולות Enqueue ו- PEnqueue צריך להיות $\Theta(1)$.
זמן W.C. של פעולת Dequeue – לא מוגבל.
זמן Amortized לפעולה צריך להיות $\Theta(1)$.

תאור המבנה:
שתי מחסניות A,B.

מימוש פעולת Enqueue וניתוח זמן ריצתה:
Push למחסנית A. זמן ריצה $\Theta(1)$.

מימוש פעולת PEnqueue וניתוח זמן ריצתה:
Push למחסנית B. זמן ריצה $\Theta(1)$.

מימוש פעולת Dequeue:

אם מחסנית B לא ריקה – נבצע Pop ממנה.

אחרת – נעביר את כל האיברים ממחסנית A למחסנית B אחד אחרי השני ואז נבצע Pop מ-B.

ניתוח זמן Amortized לפעולה במבנה:

נסתכל על סדרה של N פעולות. בסדרה כזו נכנסו למבנה לכל היותר N איברים (על ידי פעולות Enqueue ו-PEnqueue).
אם איבר נכנס על ידי פעולת PEnqueue (כלומר למחסנית B) אז סה"כ מבוצעות עליו שתי פעולות לכל היותר: Push ל-B ואולי גם Pop מ-B.
אם איבר נכנס על ידי פעולת Enqueue (כלומר למחסנית A) אז סה"כ מבוצעות עליו ארבע פעולות לכל היותר: Push ל-A ואולי גם Pop מ-A ו-Push ל-B ואולי Pop מ-B.
כלומר בכל מקרה על כל איבר "מבזבזים" זמן O(1).
לכן בכל סדרה של N פעולות זמן הריצה הוא O(N). ברור שזמן הריצה הוא גם $\Omega(N)$, כי כל פעולה בסדרה לוקחת $\Omega(1)$.
לכן כל סדרה, ובפרט בסדרה הגרועה ביותר, לוקחת זמן ריצה $\Theta(N)$.

זמן Amortized לפעולה הוא לכן: $\frac{\Theta(N)}{N}=\Theta(1)$.

מספר מחברת:

ב. (9 נקודות) נתון מבנה נתונים דינאמי המאפשר Insert, Delete ו-FindMin.

תנו חסם תחתון לזמן Amortized לפעולה, בהנחה שהמבנה מתאים לשימוש במודל ההשוואות

החסם: $\Omega(\lg N)$

הוכחה: יהי A מבנה נתונים דינמי נראה סדרה של פעולות שמאפשרת למיין מערך בגודל n:

1. נבצע n פעולות Insert.
2. נחזור n פעמים על הפעולה הבאה:

   נבצע FindMin, נדפיס את האיבר שמצאנו ונמחק אותו מהמבנה.

הנ"ל הוא אלגוריתם מיון, וברור שהוא במודל ההשוואות (כי המבנה במודל ההשוואות).

מספר תעודת זהות: [ ] 1 מועד ב', גירסה 1

מספר מחברת: [ ]

מספר קורס: 0368.2158

סמסטר א'+ב', מועד ב', תשס"ה

כ"ח אלול תשס"ה, 2/10/2005

מבחן במבני נתונים

פרופ' חנוך לוי, פרופ' טובה מילוא, עודד שוורץ, אסף שפירא ואלעד ורבין

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

הוראות כלליות:

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף בטופס זה.
2. יש לענות על כל השאלות.
3. הסברים לשאלות האמריקאיות – רשות.
4. **תשובות לשאלות הפתוחות והסברים לשאלות האמריקאיות יש לכתוב על טופס המבחן בלבד.** המחברות הן לטיוטה בלבד ולא תיבדקנה.
5. **תשובות סופיות לשאלות האמריקאיות יש לכתוב אך ורק על הטופס המצורף**. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. **ניתן וצריך** להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. במבחן 100 נקודות. ניקוד פרטני ליד השאלות.
8. **הערה חשובה לשאלות הפתוחות:** כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות/ ארוכות/ מסובכות/ מסורבלות יקבלו ניקוד חלקי **גם אם הן נכונות**. כמו כן, **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.

בהצלחה!!

תשובות לשאלות האמריקאיות:

שאלה	תשובה (הקיפו בעיגול ורשמו את האות)	
0 (דוגמה)	א ב ג ד ה ו	ג
1	א ב ג ד ה ו	
2	א ב ג ד ה ו	
3	א ב ג ד ה ו	
4	א ב ג ד ה ו	
5	א ב ג ד ה ו	
6	א ב ג ד ה ו	
7	א ב ג ד ה ו	

מספר מחברת: ______

1. (8) נתון מערך המייצג ערימת מקסימום בגודל n. מפעילים עליו את מיון QuickSort. איבר הציר בכל שלב נבחר להיות הראשון בתת המערך עליו מופעל האלגוריתם. זמן הריצה .W.C הוא:

א. $\Theta(n)$

ב. $\Theta(n\log n)$

ג. $\Theta(n^2)$

ד. $\Theta(n^2\log n)$

ה. $\Theta(n^3)$.

הסבר קצר (רשות):

2. (7) נסתכל על אלגוריתם quicksort המתבצע על מערך Aשבו מציאת הפיבוט מתבצעת כדלקמן: מצא את ערכי המינימום (min) ומקסימום(max) ב-A ובחר את הפיבוט להיות הממוצע של min ו max. במידה והאלגוריתם מתבצע על מערך שבו נמצאים האברים 1,2,..,N(בסדר שרירותי) אזי סבוכיות quicksort הינה:

א. worst case=$\Theta(n\log n)$ ו- average case= $\Theta(n\log n)$.

ב. worst case=$\Theta(n^2)$ ו- average case= $\Theta(n\log n)$.

ג. worst case=$\Theta(n^2)$ ו- average case= $\Theta(n^2)$.

ד. worst case לא חסום ו- average= $\Theta(n\log n)$

ה. אף אחת מהתשובות לעיל אינה נכונה.

הסבר קצר (רשות):

3. (7) תהא T(n) פונקציה המקיימת:

$$T(n)=T(cn)+T(dn)+n$$
$$c>0, d>0$$
$$c+d<1$$

$$T(n)=?$$

א. $\Theta(\lg n)$.

ב. $\Theta(\max\{n^c, n^d\})$.

ג. $\Theta(n)$.

ד. $\Theta(n\lg n)$

ה. $\Theta(n^{\frac{1}{c+d}})$

ו. אף אחד מהנ"ל.

הסבר קצר (רשות):

4. (7) נתון עץ חיפוש בינרי (לא בהכרח מאוזן) בגודל n ובגובה h, שבו לכל צומת יש שדה Su המצביע לעוקב, אבל השדה הנ"ל שגוי. רוצים לתקן את העץ, כלומר לתקן את השדה Su של כל הצמתים בזמן .W.C מינימלי. ניתן לעשות זאת בזמן

א. $\Theta(\lg n)$

ב. $\Theta(n)$

ג. $\Theta(n\lg n)$

ד. $\Theta(n^2)$

ה. $\Theta(nh^2)$

ו. אף אחד מהנ"ל

הסבר קצר (רשות):

5. (7) נתון מבנה נתונים A.

טענה:

אם נמחק מ-A קודם את הערך 3 ואז נמחק את הערך 4 נקבל את אותו מבנה בדיוק כמו במקרה שבו נמחק מ-A קודם את הערך 4 ואחר-כך את הערך 3 (הניחו שכל ערך מופיע לכל היותר פעם אחת).

הטענה נכונה תמיד כאשר A הוא:

א. עץ חיפוש בינרי.

ב. טבלת Hash שבה התנגשויות נפתרות ע"י רשימות מקושרות.

ג. א+ב

ד. אף אחד מהנ"ל.

הסבר קצר (רשות):

6. (7) טענה: אם $f(n)=O(g(n))$ אזי $f(n)^{1/2}=O(g(n)^{1/2})$.

א. הטענה תמיד לא נכונה

ב. הטענה תמיד נכונה

ג. הטענה לפעמים נכונה ולפעמים לא נכונה

הסבר קצר (רשות):

7. (7) טענה: אם נסייר על ערימת מינימום (Min-Heap) שכל אבריה שונים זה מזה בסיור LRD (post-order) אזי נקבל סדרה יורדת של אברי הערימה

א. הטענה תמיד לא נכונה

ב. הטענה תמיד נכונה

ג. הטענה לפעמים נכונה ולפעמים לא נכונה

הסבר קצר (רשות):

8. (30) נתון מערך בגודל n, המחולק ל-n/k קטעים באורך k כל אחד. האיברים בכל קטע גדולים מכל האיברים בקטע משמאלו וקטנים מכל האיברים בקטע מימינו. נתעניין באלגוריתם מיון למערך כזה, המתאים למודל ההשוואות, וכן בחסם תחתון לאלגוריתמי מיון במודל ההשוואות למערך כזה.

א.(12)

תאור אלגוריתם המיון:

ניתוח זמן הריצה של האלגוריתם (כפונקציה של n ו-k):

הסבירו למה האלגוריתם שהצעתם מתאים לשימוש במודל ההשוואות

ב. (12) הציעו חסם תחתון במודל ההשוואות (כפונקציה של n ו-k) הדוק ככל האפשר והוכיחו אותו.

החסם:

הוכחה:

ג. (6) הציעו אלגוריתם יעיל יותר, אם ידוע שבכל קטע, מספר הערכים השונים הוא לכל היותר lglg k:

האלגוריתם:

ניתוח זמן ריצה:

מספר מחברת:

9. (20)

א. (13) נתונה קבוצה S של n מספרים שונים. תארו אלגוריתם העובד ב-$O(n)$ זמן כך שבהינתן מספר חיובי k ($k<n$) האלגוריתם מחזיר את k המספרים הכי קרובים **לחציון** של S.

האלגוריתם:

ניתוח זמן ריצה:

ב. (7) אלגוריתם מיון נקרא יציב (stable) אם לכל שני מקומות במערך A: A[i], A[j] שהמפתחות שלהם שווים, מתקיים שהם נשארים אחרי המיון בסדר בו היו לפני המיון. נתון אלגוריתם מיון ALG שרץ בזמן f(n). הראו כיצד ניצן ליצור ממנו אלגוריתם מיון יציב ALG2 שרץ בזמן f(n)+O(n).

הפתרון:

ניתוח זמן ריצה:

פתרון למבחן:

Question:

1	2	3	4	5	6	7

Weight:

8	7	7	7	7	7	7

Version 1 - answers:

3	1	3	2	2	2	3

Matching question of Version 2:

3	7	2	1	4	5	6

E.g. question 1 of version 1 appears as question 3 in version 2.

ו
ג

1.7.2005
כד׳ סיון תשס״ה
מספר קורס: 0368.2158

מבחן במבני נתונים

סמסטר ב׳ מועד א׳

פרופ׳ טובה מילוא ואלעד ורבין

משך המבחן: שלוש שעות.

חומר סגור, ללא שימוש במחשבונים.

מותר שימוש בדף עזר אחד A4 דו צדדי.

נא לכתוב מספר מחברת ו-ת.ז. בראש כל עמוד.

תשובות לשאלות יש לתת בטופס זה בלבד.

מחברת הבחינה תשמש לטיוטה בלבד ולא תיבדק.

בכל שאלה אפשר להסתמך על נכונות סעיפים קודמים, גם אם לא פתרת אותם.

מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.

כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות/ ארוכות/ מסובכות/ מסורבלות יקבלו ניקוד חלקי **גם אם הן נכונות.**

אין חובה לכתוב אלגוריתמים בפסאודו-קוד. בכל מקרה יש לתת הסבר לרעיון הכללי.

כל הלוגריתמים בבסיס 2 אלא אם נאמר אחרת.

בהצלחה

שאלה	ניקוד
1-7	60
8	20
9	20
סה״כ	100

תעודת זהות:

מספר מחברת:

(8).1

פתרו את נוסחת הנסיגה הבאה:

$$T(n) = T(n-1) + \lg n$$

א. $T(n) = \Theta(\lg n)$

ב. $T(n) = \Theta(n)$

ג. $T(n) = \Theta(n \lg n)$

ד. $T(n) = \Theta(n \lg^2 n)$

ה. $T(n) = \Theta(n^2)$

הסבר קצר (רשות):

$T(n) = \log(n!) = \Theta(n \log n)$

(8) .2

מהו היחס האסימפטוטי בין הפונקציות הבאות: o, ω, Ω, O או Θ?

אם יש כמה תשובות נכונות, בחרו את ההדוקה ביותר.

$$f(n) = n^{\lg n}, g(n) = 4^{\lg^2 n}$$

א. $f(n) = O\left(g(n)\right)$

ב. $f(n) = \Omega\left(g(n)\right)$

ג. $f(n) = \Theta\left(g(n)\right)$

ד. $f(n) = o\left(g(n)\right)$

ה. $f(n) = \omega\left(g(n)\right)$

הסבר קצר (רשות):

$f(n) = 2^{\log^2 n} = o\left(4^{\log^2 n}\right) = o(g(n))$

תעודת זהות:

מספר מחברת:

(8) .3

רוצים למיין מערך בן n איברים שלמים שערכיהם בין 1 ל- $n^{\lg\lg n}$. האלגוריתם הכי טוב למשימה זאת רץ בזמן:

א. $T(n)=\Theta(n)$

ב. $T(n)=\Theta(n\lg\lg n)$

ג. $T(n)=\Theta(n(\lg\lg n)^2)$

ד. $T(n)=\Theta(n\lg n)$

הסבר קצר (רשות):

בסיס – מיון Radix בבסיס n.
יש $\log\log n$ "ספרות".

(8) .4

נגדיר מודל חישובי חדש בשם log-move ונרצה למיין בו מערך בגודל n. במודל log-move מותר לבצע שינוי במערך רק בעזרת פעולה של החלפה בין שני איברים שהם במרחק לכל היותר lgn זה מזה, ופעולה זו לוקחת O(1) זמן. אלגוריתם המיון המהיר ביותר במודל החדש רץ בזמן:

א. $T(n)=\Theta(n)$

ב. $T(n)=\Theta(n\lg n)$

ג. $T(n)=\Theta(\frac{n^2}{\lg^2 n})$

ד. $T(n)=\Theta(\frac{n^2}{\lg n})$

ה. $T(n)=\Theta(n^2)$

הסבר קצר (רשות):

חסם תחתון: עבור המערך ההפוך $[n, n-1, \ldots, 2, 1]$
צריך [illegible] הכלה $\log n\left(1+2+\cdots+\frac{n}{\log n}\right)\approx\frac{n^2}{\log n}$

חסם עליון: QuickSort עם הזזה בקפיצות $\log(n)$

$T(n)\le 2T(\frac{n}{2})+n\cdot\frac{n}{\log n}$
$=\Theta(\frac{n^2}{\log n})$

תעודת זהות :

מספר מחברת :

5. (9)

רוצים אלגוריתם שבהינתן מערך בן n איברים ובהינתן מספר k, מדפיס את k האיברים הגדולים ביותר במערך. מהי סיבוכיות הזמן של האלגוריתם הטוב ביותר שפותר בעיה זאת?

א. $T(n)=\Theta(k)$

ב. $T(n)=\Theta(n)$

ג. $T(n)=\Theta(n\lg k)$

ד. $T(n)=\Theta(\min(nk, n\lg n))$

הסבר קצר (רשות) :

מציאת $O(n)$ כ- $O\Omega(n-k)$

סטטיסטי k במערך והדפסת [illegible] כל [illegible] שגדולים ממנו $\Theta(n)$

$\Omega(n)$ כי חייבים לעבור על כל האיברים.

6. (9)

נתונה MIN-Heap בת 12 איברים, שכולם גדולים מ-20, ושונים. מכניסים ל-Heap את המספר 5.

עבור כמה קודקודים ב-Heap, לא כולל הקודקוד החדש, ישתנה הערך השמור בהם?

תשובה מספרית (לא נוסחה) :

3

הסבר קצר (רשות) :

דרך המסלול של 5 שווה לגובה מהעלה

עד שהוא יהיה בשורש הערימה.

תעודת זהות:

מספר מחברת:

7. (9)

רוצים מבנה נתונים ששומר מספרים שלמים ותומך בפעולות:

$INSERT(x)$ - מכניס את x למבנה

$DELETE(x)$ - מוציא את x מהמבנה

$MIN-GAP()$ - מחזיר x ו-y שמביאים למינימום את הביטוי $|x-y|$

מבין האפשרויות הבאות, בחר את הביצועים הטובים ביותר שניתן להשיג:

א.	$MIN-GAP:O(n)$	$DELETE:O(\lg n)$	$INSERT:O(\lg n)$
ג.	$MIN-GAP:O(1)$	$DELETE:O(\lg n)$	$INSERT:O(\lg n)$
ד.	$MIN-GAP:O(1)$	$DELETE:O(1)$	$INSERT:O(1)$

ה. לא ניתן להשיג אף אחת מהנ"ל

הסבר קצר (רשות):

מחזיקים עץ 2-3 ממוין כאשר בכל node

מרק מינ' בין הבן הימני הקטן שלו, הבן הגדול.

והבן הכי קטן.

הכנסה ומחיקה - $O(\log n)$ min-gap $O(1)$ כי זה רק חיפוש.

העדכון אחרי השינויים זולים בגובה.

בזמן הכנסה ומחיקה.

תעודת זהות:

מספר מחברת:

8. (20)

מגדירים אלגוריתם בשם SillyQuickSort, שהוא דומה לאלג' QuickSort כפי שלמדנו בכיתה שרץ על מערך A בגודל n שכל אבריו שונים. SillyQuickSort רץ כך שבכל שלב אי-זוגי בוחרים את ה-pivot הכי טוב (כלומר, את החציון), ובכל שלב זוגי בוחרים pivot גרוע -- את האיבר הרביעי בגודלו. הנה פסאודו-קוד של SillyQuickSort, המניח היכרות עם האלג' QuickSort:

<u>SillyQuickSort(A)</u>

Find x, the median of A

Pick x as your pivot

Partition according to x, as in QuickSort

Run SillyQuickSort2 on both parts

<u>SillyQuickSort2(A)</u>

If A is of size < 4, sort A using Insertion Sort, and return

Find x, the fourth biggest element in A

Pick x as your pivot

Partition according to x, as in QuickSort

Run SillyQuickSort on both parts

מהו זמן הריצה של SillyQuickSort ב-worst case?

זמן הריצה:	$\Theta(n \log n)$

תעודת זהות:

מספר מחברת:

הסבר:

$$T(n) = 2T'\left(\frac{n}{2}\right) + n = 2\left[T\left(\frac{n}{2}-4\right) + T(4) + \frac{n}{2}\right] + n$$

$$= 2T\left(\frac{n}{2}-4\right) + 2n + c \le 2T\left(\frac{n}{2}\right) + 3n \underset{\text{לפי המאסטר}}{=} \Theta(n\log n)$$

T - סיבוכיות של SillyQuickSort

T' סיבוכיות של SillyQuickSort2

במודל השוואות

מכיוון שהחסם למיון בדרך הוא חסם תחתון במקרה ע״י $\Omega(n\log n)$ והראנו שהוא חסם עליון במקרה ע״י $O(n\log n)$

לכן סה״כ $\Theta(n\log n)$

תעודת זהות:

מספר מחברת:

9. **(20)**

רוצים אלגוריתם יעיל ככל הניתן לבעיה הבאה:

קלט: n קטעים סגורים על הישר הנתונים בעזרת נקודות ההתחלה והסיום שלהם, כלומר בצורה $[a_1,b_1],\ldots,[a_n,b_n]$. נתון שאין נקודת קצה שמופיעה פעמיים. (שים לב: הקטעים לא בהכרח זרים!). תן אלג' יעיל שבודק האם יש בקלט שני קטעים כך שהאחד מוכל בשני.

נדרש תיאור של האלגוריתם (מותר לכתוב פסאודו-קוד **מוסבר היטב**), ונדרשת גם הוכחת נכונות מפורטת. **חלק מהניקוד יינתן על הוכחת הנכונות.**

סיבוכיות:

$\Theta(n\log n)$

תיאור האלגוריתם והוכחת נכונות:

נבנה שני עצי 2-3 של הקטעים. זמן $\Theta(n\log n)$

T_1 – ממוין לפי נק' ההתחלה a_i

T_2 – ממוין לפי נק' הסיום b_i

נעבור על T_1 בדרך ממוינת לפי איבר (n)

– נמחק את הקטע הנוכחי מעץ T_2 $\Theta(\log n)$

– נבדוק אם האיבר הקטן ביותר ב-T_2 $\Theta(\log n)$

יש לו קטע נגמר ~~לפני~~ לפני הקטע הנוכחי

הם מוכלים

סה"כ $\Theta(n\cdot\log n)$

בכל רגע נתון, (שכבודקים) (a_i,b_i), ב-T_2

יש רק קטעים שמתחילים אחרי a_i ולכן

מספיק לבדוק קטע שנגמר לפני b_i.

תעודת זהות:

מספר מחברת:

תיאור האלגוריתם והוכחת נכונות (המשך):

חסם תחתון: נניח שלה יש אלגוריתם ב- $o(n\log n)$

אזי, ניקח מערך A בגודל n של מספרים

שלמים $[x_1 \cdots x_n]$. נייצר ממנו קטעים

ש- i קטע שונו

$[\cdots[x_i - \frac{i+1}{3n}, x_i + \frac{i+1}{3n}]\cdots]$

ונפעיל את האלגוריתם שלנו ופתרנו

בעיית ED ב- $o(n\log n)$ וזה לא אפשרי.

* ישנו הקטעים מורכים אזי קיים קטע שמוכל

קיים קטע אחד שמתחיל ומסתיים ונגמר לפניו.

לכן בדיקה מי נגמר הכי מוקדם בקטעים שמתחילים

אחרי הקטע שבודקים מספיקה.

* סדר סיבוכיות

$$T(n) = \underbrace{2n\log n}_{\text{בניית עץ}} + n\cdot[\log n + \log n] = \Theta(n\log n)$$

מציאת T_1 מחיקה מ- T_2 חיפוש ב- T_2

סוף

מספר קורס: 0368.2158

סמסטר א׳, מועד א׳, תשס״ה

18/2/2005

מבחן במבני נתונים

פרופ׳ חנוך לוי, עודד שוורץ ואסף שפירא

משך המבחן שלוש וחצי שעות (אין הארכה)
הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף בטופס זה.
2. יש לענות על כל השאלות.
3. הסברים לשאלות האמריקאיות – רשות.
4. **תשובות לשאלות הפתוחות והסברים לשאלות האמריקאיות יש לכתוב <u>על טופס המבחן בלבד</u>.** המחברות הן לטיוטה בלבד ולא תיבדקנה.
5. **תשובות סופיות לשאלות האמריקאיות יש לכתוב אך ורק <u>על הטופס המצורף</u>.** בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. **<u>ניתן וצריך</u>** להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. במבחן 100 נקודות. ניקוד פרטני ליד השאלות.
8. **<u>הערה חשובה לשאלות הפתוחות:</u>** כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות/ ארוכות/ מסובכות/ מסורבלות יקבלו ניקוד חלקי **גם אם הן נכונות**. כמו כן, **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.

בהצלחה!!

מספר מחברת :

1. **(7)** נתון מערך A ובו n נתונים. בשאלה זו המשמעות של החלפה היא הפעולה Swap(i,j), כלומר החלפת איבר במקום כלשהו i במערך עם איבר המופיע במקום אחר j . נניח רוצים לתכנן אלגוריתם במודל ההשוואות שיבצע מספר קטן ככל האפשר של **החלפות במערך** (מספר ההשוואות **אינו מעניין** במקרה זה). מספר ההחלפות אותן מבצע האלגוריתם הטוב ביותר במקרה הגרוע ביותר הוא?

א. $\Theta(n)$

ב. $\Theta(n\log n)$

ג. $\Theta(n^2)$

ד. $\Theta(\sqrt{n})$

ה. $\Theta(\log n)$.

הסבר קצר (רשות) :

א. Selection sort

2. **(8)** תהא $T(n)$ פונקציה המקיימת : $T(n)=T(\log n)+1$. נגדיר $A(n)=\log(T(n))$ ו $B(n)=T(\log n)$. מה מהבאים נכון?

א. $A=\Theta(B)$.

ב. $A=o(B)$.

ג. $A=\omega(B)$.

ד. התשובה תלויה ב-n.

ה. אף אחד מהנ"ל.

הסבר קצר (רשות) :

ב. כי A(n)=log(T(n)) B(n)=T(n)-1

3. (9) השאלה הבאה מתייחסת לאלגוריתם המיון הבא המקבל מערך בגודל n (עם איברים $a_1, ..., a_n$):

1. הגרל שני אינדקסים i,j $i \neq j$ בהסתברות אחידה מהמספרים 1,...,n. נניח בלי הגבלת הכלליות כי $a_i<a_j$.
2. חלק את המערך לשלושה חלקים כך שבחלקו השמאלי יהיו האיברים הקטנים מ a_i, בחלקו הימני האיברים הגדולים-שווים מ a_j ובחלקו האמצעי האיברים שקטנים מ a_j וגדולים שווים מ a_i.
3. הפעל את האלגוריתם רקורסיבית על כל אחד משלושת החלקים.

<u>שימו לב:</u> את השלב השני ניתן לבצע ב O(n) בדומה לפעולת ה Partition של QuickSort.

מהי <u>תוחלת זמן הריצה</u> של האלגוריתם?

א. $\Theta(n)$

ב. $\Theta(n\log n)$

ג. $\Theta(n^2)$

ד. $\Theta(n^2 \log n)$

ה. $\Theta(n^{\log_2 3})$.

הסבר קצר (רשות):
ב. זהו quicksort (למה?)

4. (9) מהו הגובה של עץ ההחלטות של אלגוריתם המיון HeapSort על מערך בגודל n?

א. $\Theta(n)$

ב. $\Theta(n\log n)$

ג. $\Theta(n^2)$

ד. $\Theta(n!)$

ה. $\Theta(n!\lg n)$

ו. אלגוריתם המיון HeapSort לא מתאים למודל ההשוואות ולכן השאלה חסרת משמעות.

הסבר קצר (רשות):
ב. כי זוהי סבוכיות ה worst case של מספר ההשוואות באלגוריתם זה.

5. (8) בערימת מקסימום, החציון נמצא בהכרח:

א. בשורש.

ב. בעומק לכל היותר $\frac{\lg(n)}{2}-1$.

ג. בעומק לכל היותר $\frac{\lg(n)}{2}+1$

ד. בשתי השכבות הנמוכות ביותר.

ה. אף אחד מהנ"ל.

הסבר קצר (רשות): ה. החציון יכול להיות בן ישיר של השורש (למשל אם כל תת עץ ימין גדולים מכל תת עץ שמאל) ויכול גם להיות באחד העלים (אם חצי הערכים הקטנים ביותר נמצאים בעלים).

6. (8) יהא T עץ AVL בעל n צמתים. גבהו של צמת מוגדר כאורך המסלול (בקשתות) הארוך ביותר ממנו לעלה. לדוגמה – גבהו של עלה הינו 0. יהא m הצומת שמכיל את ערך המינימום בעץ. אזי גבהו של m ב-T הינו:

א. בהכרח 0.

ב. קטן מ או שווה ל 1.

ג. בדיוק 1.

ד. קטו מ או שווה ל 2.

ה. יכול להיות $\Theta(\log n)$.

ו. יכול להיות $\Theta(n)$.

הסבר קצר (רשות):
ב. למינימום אין בן שמאלי. לכן הוא עלה או צומת עם בן ימני שחייב, לפי הגדרת AVL , להיות עלה.

7. (9) נתון מערך A ובו n נתונים שונים זה מזה (**לא** ממוינים). נתונה ערימת מינימום H ובה אותם נתונים. מעוניינים לקחת את $\sqrt{n}$ האיברים הקטנים ביותר ולמיינם בזמן Worst case הנמוך ביותר. יהא a(n) עלות ביצוע הפעולה (על ידי האלגוריתם הטוב ביותר שהנכם מכירים) על A ותהא h(n) עלות היבצוע (על ידי האלגוריתם הטוב ביותר שהנכם מכירים) על H. אזי:

א. עלות הבצוע של שני האלגורתמים זהה.

ב. עלות הבצוע של שני האלגורתמים **בסדר גודל** זהה אבל הקבוע של a(n) נמוך יותר.

ג. עלות הבצוע של שני האלגורתמים **בסדר גודל** זהה אבל הקבוע של h(n) נמוך יותר.

ד. a(n)=o(h(n)).

ה. h(n)=o(a(n)).

הסבר קצר (רשות): ה. על A מבצעים ORDER STAT ואחכ מיון. עלות: $\Theta\left(n+1/2\sqrt{n}\log n\right)=\Theta\left(n\right)$ על H מבצעים $\sqrt{n}$ DELETEMIN . עלות: $\Theta\left(\sqrt{n}\log n\right)$

I (15) תארו מבנה נתונים המתחזק קבוצה של נקודות במישור, כאשר כל נקודה p_i מיוצגת כזוג מספרים (x_i,y_i). מבנה הנתונים צריך לתמוך בפעולות insert(x,y), delete(x,y), find(x,y) בזמן גרוע ביותר O(log n), וכמו כן תומך בפעולה $CountPointsAtDistance(d_1,d_2)$, המבצעת את הפעולה הבאה: בהינתן שני מספרים d_1,d_2 מחזירה הפעולה את **מספר הנקודות** במבנה שמרחקן מהראשית גדול מ d_1 וקטן מ d_2 (זיכרו כי מרחק נקודה (x,y) מהראשית הוא $\sqrt{x^2+y^2}$).

$CountPointsAtDistance(d_1,d_2)$ צריכה להתבצע בסיבוכיות הנמוכה ביותר האפשרית.

(7) תאור המבנה

עץ חיפוש מאוזן (2-3 או AVL למשל) שבו המפתח הינו המרחק d. מתחת לכל עלה מוחזק עץ 2-3 שבו המפתחות המתאימים (x,y) בסדר לקסיקוגרפי.
העץ הראשי בנוי כעץ ORDER STAT בו בכל צומת שומרים את מספר המפתחות שנמצאים בתת העץ.

(8) כיצד ממומשת הפעולה CountPointsAtDistance ומה הסיבוכיות שלה:

מוצאים את הנקודה הרחוקה ביותר מהראשית שמרחקה קטן מ-d2.
מחשבים את הסדר הסטטיסטי שלה: a.
מוצאים את הנקודה הקרובה ביותר לראשית שמרחקה גדול מ-d2.
מחשבים את הסדר הסטטיסטי שלה: b.
מחזירים את a-b+1.
כל אחת מהפעולות הנ"ל מבוצעת בזמן W.C. O(log n) לכן זהו זמן ריצת הפעולה.

II (15) יהא A אלגוריתם מיון כלשהו. עבור מספר כלשהו n, נגדיר את Compare(n,i) להיות מספר הפעמים הגדול ביותר (על פני כל הקלטים בגודל n) ש A משווה את האיבר שנמצא במקום i בקלט עם איבר כלשהו אחר. כמו כן נגדיר את Compare(n) להיות המקסימום על פני כל i של Compare(n,i).

א. (5) עבור MergeSort, האם Compare(n)=O(log n)? הסבירו.

$\Theta(n)$. בפאזה האחרונה אם מערך אחד מכיל אברים גדולים מהשני, מוביל לעלות הזו.

עבור קלט מסוים I ואיבר מסוים x בקלט נגדיר Compare(n,x,I) להיות מספר הפעמים הגדול ביותר ש A משווה את x עם איבר כלשהו אחר.

ב. (4)
האם ניתן לכתוב אלגוריתם מיון חדש B שמקבל גם את x בקלט, כך ש Compare(n,x,I)= O(logn) ?
הסבירו.

כן. מיון שאר האיברים ובצוע binary search ל-x.

ג. (6) האם יש אלגוריתם מיון במודל השוואות המקיים Compare(n)=O(log log n)?

לא. זה יאפשר מיון ב O(nlglgn) השוואות. זמן הריצה אמנם יכול להיות גדול יותר, אבל הראינו שמספר ההשוואות לאלגוריתם מיון במודל זה הוא $\Omega(n \lg n)$.

III: **(12)** תארו אלגוריתם לפתרון הבעיה הבאה:

קלט: קבוצה כלשהי $A=\{a_1,...,a_n\}$ של n מספרים.

מטרה: האלגוריתם צריך לבנות **בזמן ממוצע טוב ביותר** מבנה נתונים DS בעל התכונה הבאה: בהינתן אוסף S של m מספרים שאיבריו לקוחים מ A, ניתן להיעזר ב DS בכדי למיין את איברי S בזמן O(max(m,n)) **ב WC**. שימו לב כי S יכולה להכיל איבר כלשהו של A **מספר פעמים**. על כן ייתכן כי מתקיים m>n.

תאור מבנה הנתונים DS:

Perfect hash של האיברים עבור ערכם. לכל איבר שומרים גם את ה rank שלו (הסדר הסטטיסטי).

תאור בניית המבנה:

ממיינים המערך –O(nlgn) – למציאת rank של כל איבר.
מכניסים את האיברים לperfecct hash O(n) בממוצע.

זמן בניית המבנה:

כיצד ניתן למיין בעזרת מבנה הנתונים:

מוצאים את הסדר הסטטיסטי של כל איבר ב-S. ממיינים את S לפי הסדרים הסטטיסטים בעזרת CountSort.

סוף

תעודת זהות:

מספר מחברת:

21.11.04

ח׳ כסלו, תשס״ה

מבחן במבני נתונים

סמסטר קיץ 2004 מועד ב׳ ומועד מיוחד סמסטר א+ב

פרופ׳ חנוך לוי, עודד שוורץ, אסף שפירא, אלעד ורבין

משך המבחן: שלוש שעות.

חומר סגור, ללא שימוש במחשבונים.

מותר שימוש בדף עזר אחד A4 דו צדדי.

נא לכתוב מספר מחברת ו-ת.ז. בראש כל עמוד.

תשובות לשאלות יש לתת בטופס זה בלבד.

מחברת הבחינה תשמש לטיוטה בלבד ולא תיבדק.

במבחן 4 שאלות. כל שאלה 25 נקודות.

בכל שאלה אפשר להסתמך על נכונות סעיפים קודמים, גם אם לא פתרת אותם.

מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.

כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות/ ארוכות/ מסובכות/ מסורבלות יקבלו ניקוד חלקי **גם אם הן נכונות.**

אין חובה לכתוב אלגוריתמים בפסאודו-קוד. בכל מקרה יש לתת הסבר לרעיון הכללי.

בהצלחה

שאלה	ניקוד
1	
2	
3	
4	
סה״כ	

תעודת זהות:

מספר מחברת:

1. נתון מערך בגודל n שבו ה-OrderStatistic(i) (האיבר ה-i בגודלו) נמצא באחד המקומות
.i-k,i-k+1,….,i+k-1,i+k

א. מצאו אלגוריתם יעיל ככל הניתן **למיון** מערך זה כאשר k = lg n.

סיבוכיות האלגוריתם (כפונקציה של n):

תיאור האלגוריתם:

תעודת זהות:

מספר מחברת:

ב.

מצאו אלגוריתם יעיל ככל הניתן **לחיפוש** איבר במערך זה, כאשר $k=\lg n$

סיבוכיות האלגוריתם (כפונקציה של n):

תיאור האלגוריתם:

תעודת זהות:

מספר מחברת:

.2

א.

פתרו את נוסחת הנסיגה הבאה:

$$T(n)=\sqrt{n}T\left(\sqrt{n}\right)+n$$

פתרון: $T(n)=\Theta(\_\_\_\_\_\_)$

הוכחה:

תעודת זהות:

מספר מחברת:

ב.

מהו היחס האסימפטוטי בין הפונקציות הבאות: o, ω, Ω, O או Θ?

$$g(n) = n \cdot \sum_{i=1}^{n} \frac{1}{2^i} \qquad f(n) = n \cdot \sum_{i=1}^{\lg n} \frac{i}{2^i}$$

היחס הוא: $f(n) = $ _____ $(g(n))$

הוכחה:

תעודת זהות :

מספר מחברת :

.3

א.

טענה :

מספר העלים בעץ בינרי הוא בדיוק אחד ועוד מספר הצמתים שיש להם שני בנים.

הטענה : נכונה לא נכונה (מחקו את המיותר).

הוכחה :

ב.

צ״ל : גובה של עץ AVL עם n צמתים הוא $\Theta\left(\lg n\right)$.

תזכורת : עץ AVL הוא עץ שבו לכל צומת ההפרש בין גובה תת עץ ימין לגובה תת עץ שמאל הוא 0, 1 או 1-.

הוכחה :

תעודת זהות:

מספר מחברת:

4.

א.

הציעו אלגוריתם **לא רקורסיבי** שמדפיס את כל המפתחות של עץ חיפוש בינרי בסדר Inorder (כלומר סדר LDR). זכרון מותר לשימוש: O(1). אסור לשנות את העץ (גם לא זמנית). האלגוריתם צריך להיות יעיל ככל האפשר. הסבירו במפורש מה מבצעים בכל שלב.

האלגוריתם:

זמן ריצה .W.C:

הוכחה:

תעודת זהות :

מספר מחברת :

ב.

טענה :

לכל מערך A קיים עץ חיפוש **בינרי** T כך ש-A הוא סריקת DLR (סריקת PreOrder) של T.

הטענה : נכונה לא נכונה (מחקו את המיותר).

הוכחה :

מועד א׳, גירסה 1

מספר תעודת זהות:

מספר מחברת:

מספר קורס: 0368.2158

סמסטר ב׳, מועד א׳, תשס״ד

9/7/2004

This exam Got 99

מבחן במבני נתונים

פרופ׳ חנוך לוי ואסף שפירא

משך המבחן שלוש וחצי שעות

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף בטופס זה.
2. יש לענות על כל השאלות.
3. הסברים לשאלות האמריקאיות – רשות.
4. **תשובות לשאלות הפתוחות והסברים לשאלות האמריקאיות יש לכתוב <u>על טופס המבחן בלבד</u>**. המחברות הן לטיוטה בלבד ולא תיבדקנה.
5. **תשובות סופיות לשאלות האמריקאיות יש לכתוב אך ורק <u>על הטופס המצורף</u>**. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. **<u>ניתן וצריך</u>** להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. במבחן 100 נקודות. ניקוד פרטני ליד השאלות.
8. **<u>הערה חשובה לשאלות הפתוחות:</u>** כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות/ ארוכות/ מסובכות/ מסורבלות יקבלו ניקוד חלקי **גם אם הן נכונות**. כמו כן, **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.

בהצלחה!!

מספר מחברת:

(8) .1

משנים את אלגוריתם QuickSort (גרסה שבה **<u>הפיבוט נבחר באמצעות פונקצית החציון</u>**) כך שאם המערך ממוין האלגוריתם מחזיר את המערך ולא מבצע שתי קריאות רקורסיביות נוספות. נניח מריצים את האלגוריתם החדש על מערך בו כל איבר מופיע בדיוק n/logn פעמים (לכן הוא מכיל logn איברים שונים). זמן הריצה של האלגוריתם ב WC הינו:

א. $\Theta(n)$

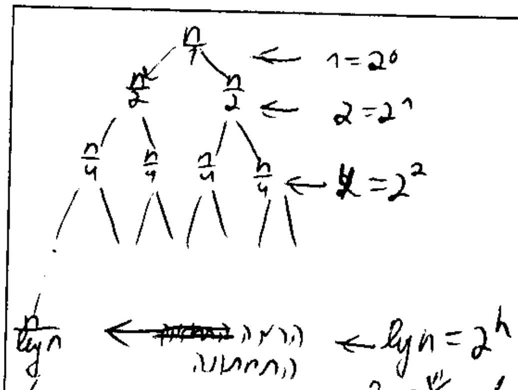
ב. $\Theta(n \log\log n)$

ג. $\Theta(n \log n)$

ד. $\Theta(n^2)$

ה. $\Theta(n \log\log n/\log n)$

הסבר קצר (רשות):



$\log\log n=\log 2^h=h$

סה״כ העץ מלא log log n רמות. כל רמה עולה $\Theta(n)$ $\Theta(n\log\log n)$.

מכאן והלאה לא ממשיכים למיין כי כל חלק כבר ממוין.

מספר תעודת זהות:

מספר מחברת:

.2 **(8)**

מה מהבאים נכון לגבי הפונקציה T המקיימת את המשוואה הרקורסיבית הבאה:

$T(1)=T(2)=1$,$T(n)=T(n-1)+T(n-2)$.

א. $T(n)=\Theta(n)$

ב. $T(n)=\Theta(n^2)$

ג. $T(n)=\Theta(n^3)$

ד. $T(n)=\Theta(n^4)$

ה. $T(n)=\Theta(n^5)$

ו. $T(n)=\Omega(n^{2004})$

הסבר קצר (רשות): זוהי פונק' בעלת סיבוכיות אקספוננציאלית, ולכן הסיבוכיות האסימפטוטית גדולה מ-n בחזקה קבועה כלשהי. לכן ו.

מספר מחברת:

(8) .3

נתונים שני עצי חפוש אדום שחור המכילים כל אחד n צמתים. הניחו שהערכים בעצים זרים זה לזה (אין ערך שמופיע פעמיים בעץ אחד ואין ערך שמופיע בשני העצים). רוצים להפוך את שני העצים לעץ לעץ חיפוש אדום שחור אחד. בהנחת מודל ההשוואות ניתן לעשות זאת בזמן (הטוב ביותר האפשרי):

א. $\Theta(n)$ W.C., $\Theta(n)$ ממוצע.

ב. $\Theta(n)$ W.C., $\Theta(n\lg n)$ ממוצע.

ג. $\Theta(n\lg n)$ W.C., $\Theta(n)$ ממוצע.

ד. $\Theta(n\lg n)$ W.C., $\Theta(n\lg n)$ ממוצע.

ה. אף אחד מהנ"ל

ו. א+ב+ג

הסבר קצר (רשות): נעשה inorder על 2 העצים ($\Theta(n)$) ל-2 מערכים (כל עץ למערך אחר) אח"כ ניצור מערך אחד ממוין משני המערכים (כמו merge) ב-$\Theta(n)$. ואז נבנה מהמערך הממוין (מכיל את כל האיברים, כמו שנאמר בתרגיל) ניתן לבנות עץ אדום-שחור ממערך ממוין ב-$\Theta(n)$. לכן סה"כ: $\Theta(n)$.
(גם ב-W.C., + ממוצע)

מספר מחברת:

.4 (4)

נניח כי פונקציה ~~f~~ $f(n)$ מקיימת: $f<10n^2$ לכל $1<n<1000$. אזי:

א. $f=O(n^2)$ בוודאות.

ב. ייתכן כי מתקיים $f=\Omega(n^{10})$.

ג. בוודאות f אינה מקיימת $f=O(n^2)$.

ד. אף תשובה שלעיל אינה נכונה.

הסבר קצר (רשות): למשל הולך כך: ~~[illegible]~~

$$f(n)=\begin{cases} n & 1<n<1000 \\ f(n-1)+f(n-2) & \text{אחרת} \end{cases}$$

$f(n)$ בעלת סיבוכיות אקספוננציאלית, ולכן במקרה זה $f(n)=\Omega(n^{10})$ (לכל גדול מ-1)

.5 (5)

נניח כי פונקציה ~~f~~ $f(n)$ מקיימת: $f>10n^2$ לכל n זוגי. אזי:

א. $f=\Omega(n^2)$ בוודאות.

ב. בוודאות f אינה מקיימת $f=o(n^2)$.

ג. בוודאות f אינה מקיימת $f=O(n^2)$.

הסבר קצר (רשות): ממש לפי הגדרה של o (o קטן). (כלומר בזוגיים זוגי לא מתקיים בוודאות).

מספר מחברת:

(8) .6

נניח כי משתמשים ברשימה מקושרת בכדי לתמוך בפעולות insert, find (אין delete). כמו כן ידוע מראש כי על כל איבר במבנה תתבצע לכל היותר פעולת find אחת. אזי זמן ה amortized של סדרה כלשהי של n פעולות המקיימות את התנאי לעיל הוא:

א. $\Theta(1)$

ב. $\Theta(\log n)$

ג. $\Theta(\log^2 n)$

ד. $\Theta(n^{1/2})$

ה. $\Theta(n)$

ו. $\Theta(n^2)$

הסבר קצר (רשות): insert לוקח $O(1)$ ולכן הסדרה הגרועה ביותר תיגרם בהכנסת כמה שיותר חיפושים. מכיוון שמחפשים כל איבר לכל היותר פעם אחת, הסדרה הגרועה ביותר היא $\frac{n}{2}$ הכנסות, ו-$\frac{n}{2}$ חיפושים (כל איבר פעם אחת). סה"כ

$$\frac{\frac{n}{2}+\sum_{i=1}^{n/2} i}{n} = \frac{\frac{n}{2}+\frac{\frac{n}{2}\left(\frac{n}{2}+1\right)}{2}}{n} = \frac{\Theta(n^2)}{\Theta(n)} = \Theta(n)$$

מספר מחברת:

(5) .7

בכיתה למדנו על HEAP בינארי. ברצוננו לממש את ה-HEAP על עץ 16-רי (לכל צומת 16 ילדים). הננו מעוניינים לבצע על העץ את פעולת **ההשקעה** של **השורש** כלפי מטה (**הפעולה שמתבצעת בפעולת** MIN **DELETE**).

יהי B(N) מספר הפעולות ב- worst case לבצוע **ההשקעה** בעץ **בינארי**.

יהי T(N) מספר הפעולות ב- worst case לבצוע **ההשקעה** בעץ **16-רי**.

הניחו שN הינו גדול ושעבור N הנתון שני העצים הינם מאוזנים לחלוטין.

א. $T(N) = B(N)$

ב. $T(N) < B(N)$

ג. $T(N) > B(N)$

הפעולות ב-2 וב-16 [illegible] מושפעות של הילדים [illegible]

הסבר קצר (רשות):

$B(N) = 2\log_2 N$

$T(N) = 16\log_{16} N = 16\frac{\log_2 N}{\log_2 16} = 16\cdot\frac{1}{4}\cdot\log_2 N = 4\cdot\log_2 N = 2B(N)$

$\Rightarrow T(N) > B(N)$

(4) .8

בנתונים של שאלה 7 הננו מעוניינים כעת בפעולת **ההצפה** של **עלה** כלפי מעלה (הפעולה שהתבצעה בפעולת INSERT).

יהי B(N) מספר הפעולות ב- worst case לבצוע **ההצפה** בעץ **בינארי**.

יהי T(N) מספר הפעולות ב- worst case לבצוע **ההצפה** בעץ **16-רי**.

א. $T(N) = B(N)$

ב. $T(N) < B(N)$

ג. $T(N) > B(N)$

[illegible] log [illegible]

הסבר קצר (רשות):

$B(N) = \log_2 N$

$T(N) = \log_{16} N = \frac{1}{4}\cdot\log_2 N = \frac{1}{4}\cdot B(N)$

$\Rightarrow T(N) < B(N)$

מספר מחברת:

9. (8) נרצה לתכנן מבנה נתונים שיתמוך בפעולות insert, delete, find עבור מספרים שלמים וכמו כן בפעולה SuccessorWithSameParity(x) המקיימת את התכונה הבאה:

i) אם x זוגי הפעולה מחזירה את המספר הזוגי הקטן מבין המספרים הזוגיים במבנה שגדולים מ x.

ii) אם x אי-זוגי הפעולה מחזירה את המספר האי-זוגי הקטן מבין המספרים האי-זוגיים במבנה שגדולים מ x.

בהנחה שרוצים לתמוך בfind בסבוכיות O(log n) WC, מה מהבאים נכון לגבי מבנה הנתונים בעל ביצועי ה WC הטובים ביותר.

<u>**תשומת לב**: x הינו ערך של מפתח ולא מצביע.</u>

א. סיבוכיות insert ו SuccessorWithSameParity היא O(1).

ב. סיבוכיות insert ו SuccessorWithSameParity היא O(log n).

ג. סיבוכיות insert היא O(log n) וסיבוכיות SuccessorWithSameParity היא O(1).

ד. סיבוכיות insert היא O(1) וסיבוכיות SuccessorWithSameParity היא O(log n).

ה. סיבוכיות insert היא O(log n) וסיבוכיות SuccessorWithSameParity היא O(n).

הסבר קצר (רשות): נשתמש ב-2 עצי 2-3 — האחד לזוגיים והאחר לאי-זוגיים. בנוסף יש רשימות מקושרות בין העלים בכל עץ כמו שעשינו בהרצאה, שזה לא פוגע בסיבוכיות האסימפטוטית.

חיפוש: נבדוק ב-O(1) באיזה עץ לחפש, ונבצע חיפוש ב-O(log n).

SuccessorWithSameParity: נגיע כמו חיפוש, אבל כשמגיעים לעלה עם x או איפה שהוא היה צריך להיות אם לא קיים — ~~נסתכל לאיבר הבא~~ נסתכל לאיבר הבא ונחזיר אותו, ואם x לא קיים נחזיר את הבא אחריו. ב-O(1) נלך לצד אחד ברשימה, ונחזיר את הערך (או NULL אם ~~[illegible]~~ הגענו לסוף הרשימה).

אם x לא נמצא, אז הגענו לעלה שלידו [illegible] להחזיר.

I. (9)

בהינתן קבוצת מספרים רוצים לבנות מבנה נתונים שיתמוך בפעולות הבאות ב O(1) ב WC:

1. find(x) המחזירה מצביע לאיבר שמפתחו x באם ישנו איבר כזה, ו Null אחרת.
2. Successor(x) מחזירה את האיבר הקטן ביותר במבנה שערכו גדול מ x.

הציעי מבנה ואת האלגוריתם לבניתו ונתחי את סבוכיותו:

יש לענות תשובה מלאה במשבצת:

$O(n \log n)$

הסבוכיות (גם בתוחלת וגם ב-worst case): באיתחול תוחלת [illegible] W.C. [illegible]

בניה ה- perf.hash בתוחלת O(n) אבל, אנחנו גם ממיינים את n המספרים בעזרת quicksort המשופר מההרצאה אז בתוחלת אנחנו מקבלים O(nlogn). [illegible]

find(x) [illegible] perfect-hash [illegible] O(1) [illegible] W.C.

successor(x) W.C ותוחלת O(1) [illegible]

תאור מבנה הנתונים: טבלת perfect-hash [illegible] את n המספרים [illegible] ממוין [illegible] per.hash [illegible] במערך הממוין.

הסבר האלגוריתם: באיתחול [illegible] את n האיברים [illegible] ($\Theta(n \log n)$ [illegible]) [illegible] quicksort המשופר [illegible] perfect-hash [illegible] O(n) [illegible] W.C. [illegible] O(1) [illegible]

find(x) – [illegible] perfect-hash [illegible] NULL [illegible] O(1).

successor(x) – [illegible] NULL [illegible] index+1 [illegible] O(1) [illegible] W.C.

מספר מחברת:

II (12) נתונים n מספרים $a_1,\dots,a_n$ מהתחום [0,2]. הניחו שלכל $i\neq j$ מתקיים $|a_i-a_j|>1/n$.

הראו שניתן למיין את המספרים בזמן O(n):

א. תארו את האלגוריתם: bucket sort ("בגימי ולי") עם 2n תאים. [illegible] ניצור מערך בגודל 2n+1.

בתא ה-i (הבא: i) יוחזק המספר שבתחום $(\frac{i}{n}, \frac{i+1}{n}]$.

מכיוון שבגלל שלתחום בגודל מעט מ-$\frac{1}{n}$, לא יתכן ש-2 מספרים a_i, a_j ($i\neq j$) ימופו לאותו תא במערך. האלגוריתם: נעבור על n המספרים, נכניס אותם למקום (לפי שהוגדר לעיל). אחרי זה, נעבור על המערך ונדפיס את האיברים שהם הערך שונה מ-(-1). זהו מיון ב-O(n).

ה. הערה: איתחול המערך בגודל 2n ע"י הצבת (-1) בכל תא — O(n).
מעבר על n המספרים ולמפות אותם לתאים המתאימים — O(1) לכל מספר, וכן לn המספרים — O(n).
לעבור על המערך ולהדפיס את האיברים השונים מ-(-1) — O(n)
סה"כ מיון — O(n).

מ.ש.ל.

מספר מחברת:

IIIא) (9) נגדיר פעולה Slice על ערמה באופן הבא: הפקודה Slice מקבלת ערמה חוקית H השמורה במערך A, ומארגנת את המערך A כך שבחציו הימני תימצא ערמה חוקית H_1 אשר תכיל את n/2 האיברים הגדולים של H וכך שבחציו השמאלי תימצא ערמה חוקיות H_2 אשר תכיל את n/2 האיברים ~~הגדולים~~ הקטנים של H. האם ניתן לממש את Slice בזמן O(n) **<u>במודל ההשוואות</u>**?

ניתן/~~לא ניתן~~:

הוכחה: במערך A שאותו מקבלים נמצא את החציון ב- O(n) (W.C.). נבצע עליו partition כלומר משמאלו יהיו כל האיברים הקטנים ממנו ומימינו יהיו כל האיברים הגדולים ממנו — O(n) ב-W.C. עכשיו על כל חצי (ימני ושמאלי בנפרד) נבצע buildheap ב- O(n) W.C. בסוף מה שקיבלנו זה מערך שבחציו השמאלי נמצאת ערימה חוקית המכילה את $\frac{n}{2}$ המספרים הקטנים של A, ובחציו הימני נמצאת ערימה חוקית המכילה את $\frac{n}{2}$ המספרים הגדולים של A, זה בדיוק מה שביקשו, וב- W.C. הכל O(n). לכן ניתן! מ.ש.ל.

9/9

מספר מחברת:

ב) (12) האם ניתן לממש את Slice מהסעיף הקודם בזמן O(logn) **במודל ההשוואות?**

~~ניתן~~/לא ניתן

הוכחה: נניח בשלילה שניתן היה לממש את Slice מהסעיף הקודם ב- O(logn). אז נמיין את המערך באופן הבא: נפעיל Slice למערך, ואז רקורסיבית נפעיל Slice על החלק הימני ועל החלק השמאלי, וכך הלאה עד שנגיע למערכים בגודל 1.

הנוסחה הרקורסיבית היא:

$$T(n) = 2T(\tfrac{n}{2}) + \log n$$

ה-Slice הראשון — קריאה רקורסיבית על כל חלקי ה-Slice..

לפי שיטת האב אנחנו מקבלים $T(n)=O(n)$. כלומר מיינו מערך בגודל n ב- (זמן) $o(n\log n)$ בסתירה לחסם התחתון $O(n\log n)$ שהוכחנו למיון במודל ההשוואות. לכן לא יתכן ש- Slice תמומש ב- $\log n$.

מ.ש.ל.

טוב. איפה [illegible] הטלק? -1

- סוף -

תעודת זהות:

מספר מחברת:

10.10.2004

כד׳ אלול תשס״ד

מספר קורס: 0368.2158

מבחן במבני נתונים

סמסטר א׳+ב׳ מועד ב׳ סמסטר קיץ מועד א׳

פרופ׳ חנוך לוי, עודד שוורץ, אסף שפירא ואלעד ורבין

משך המבחן: שלוש שעות.

חומר סגור, ללא שימוש במחשבונים.

מותר שימוש בדף עזר אחד A4 דו צדדי.

נא לכתוב מספר מחברת ו-ת.ז. בראש כל עמוד.

תשובות לשאלות יש לתת בטופס זה בלבד.

מחברת הבחינה תשמש לטיוטה בלבד ולא תיבדק.

בכל שאלה אפשר להסתמך על נכונות סעיפים קודמים, גם אם לא פתרת אותם.

מותר להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.

כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות/ ארוכות/ מסובכות/ מסורבלות יקבלו ניקוד חלקי **גם אם הן נכונות.**

אין חובה לכתוב אלגוריתמים בפסאודו-קוד. בכל מקרה יש לתת הסבר לרעיון הכללי.

בהצלחה

שאלה	ניקוד
א	
ב	
ג	
ד	
סה״כ	

תעודת זהות:

מספר מחברת:

(8).1

פתרו את נוסחת הנסיגה הבאה:

$T\left(n\right)=T(cn)+T\left(\left(1-c\right)n\right)+n$ כאשר נתון ש-c קבוע, $0<c<1$

א. $T\left(n\right)=\Theta\left(n\right)$

*ב. $T\left(n\right)=\Theta\left(n\lg n\right)$

ג. $T\left(n\right)=\Theta\left(n\sqrt{n}\right)$

ד. $T\left(n\right)=\Theta\left(n^2\right)$

ה. $T\left(n\right)=\Theta\left(n^c\right)$

הסבר קצר (רשות):
הוכחה באינדוקציה או על ידי עץ רקורסיה.

2. **(8)**

מהו היחס האסימפטוטי בין הפונקציות הבאות: o,ω,Ω,O או Θ?

אם יש כמה תשובות נכונות, בחרו את ההדוקה ביותר.

$$f(n)=\frac{n}{\lg n},\ g(n)=\frac{n^2}{\left(\lg n\right)^{2004}}$$

א. $f\left(n\right)=O\left(g\left(n\right)\right)$

ב. $f\left(n\right)=\Omega\left(g\left(n\right)\right)$

ג. $f\left(n\right)=\Theta\left(g\left(n\right)\right)$

*ד. $f\left(n\right)=o\left(g\left(n\right)\right)$

ה. $f\left(n\right)=\omega\left(g\left(n\right)\right)$

הסבר קצר (רשות):
לכל $\varepsilon>0$ *מתקיים* $\log n = o(n^{\varepsilon})$.

תעודת זהות :

מספר מחברת :

3. **(9)**

משנים את אלגוריתם MergeSort כך שאם המערך ממוין האלגוריתם מחזיר את המערך ולא מבצע שתי קריאות רקורסיביות נוספות. נניח מריצים את האלגוריתם החדש על מערך בו כל איבר מופיע בדיוק n/logn פעמים (לכן הוא מכיל logn איברים שונים). זמן הריצה של האלגוריתם ב WC הינו :

א. $\Theta(n)$

ב. $\Theta(n \log \log n)$

*ג. $\Theta(n \log n)$

ד. $\Theta(n^2)$

הסבר קצר (רשות) :
עבור הקלט הבא, זמן הריצה יהיה כמו באלגוריתם MergeSort המקורי:
מערך המורכב מ-n/logn חלקים, בכל חלק logn ערכים בסדר יורד.

4. **(8)**

נגדיר מודל חישובי חדש בו נרצה למיין מערכים של מספרים בשם Only-Adjacent-Swaps.
במודל אסור להחליף בין שני איברים כלשהם במערך,אלא רק בין זוג איברים סמוכים. אלגוריתם המיון המהיר ביותר במודל החדש רץ בזמן :

א. $\Theta(n \log n)$

ב. $\Theta(n \log^2 n)$

*ג. $\Theta(n^2)$

ד. $\Theta(n^2 \log n)$

ה. $\Theta(n^3)$

הסבר קצר (רשות) :
מערך הממוין בסדר הפוך ידרוש $\Theta(n^2)$ פעולות החלפה.

תעודת זהות :

מספר מחברת :

5. **(9)** נתון עץ Order-Statistics כפי שהוגדר בכיתה, עם השדה הנוסף size (כלומר size של צומת x הוא מספר הצמתים, כולל x, בתת העץ שזה ש-x הוא השורש שלו). בכמה זמן ניתן להדפיס את k הצמתים בעץ להם שדה size גדול ביותר. מותר להניח ש $k < \sqrt{n}$.

א. $\Theta(n)$.

ב. $\Theta(n \log n)$.

ג. $\Theta(k \log n)$.

*ד. $\Theta(k \log k)$.

ה. $\Theta(k)$.

הסבר קצר (רשות) : בדומה להדפסת k האיברים הגדולים ביותר בערימת מקסימום. נשתמש בערימת עזר, שתכיל את המועמדים להדפסה, ובכל שלב נדפיס את המקסימלי של ערימת עזר, נמחק אותו מערימת העזר, ואת שני בניו מהעץ נכניס במקומו לערימת העזר.

6. **(8)**

רוצים מבנה נתונים שתומך בפעולות Insert, Delete ו-Find.

מוצעים שני מבנים :

הראשון תומך בכל הפעולות בזמן O(n) W.C. ובזמן Amortized לפעולה O(1).

השני תומך בכל הפעולות בזמן O(log n) W.C. ובזמן Amortized לפעולה O(log N).

נזכיר ש-n מתאר את מספר האיברים במבנה, ו-N מתאר את מספר הפעולות שבוצעו על המבנה.

המבנה דרוש לאלגוריתם שמבצע N פעולות על המבנה. המבנה מאותחל למצב ריק. בתחילת ריצת האלגוריתם מבוצעות עליו $\frac{N}{10}$ פעולות insert.

איזה משני המבנים עדיף, על מנת להשיג זמן W.C. הנמוך ביותר לאלגוריתם?

*א. הראשון.

ב. השני.

ג. התשובה עבור $n = \frac{N}{2}$ שונה מהתשובה עבור $n = \frac{N}{4}$.

ד. אף אחת מהתשובות הנ"ל.

הסבר קצר (רשות) :

הראשון דורש זמן כולל O(N). השני דורש זמן כולל O(N log N).

תעודת זהות :

מספר מחברת :

7. **(9)**

נתעניין בבעיה הבאה :

קלט : מערך בגודל n.

שאלה : האם יש במערך 2 ערכים שמספר המופעים של שניהם יחד הוא **בדיוק** 2004.

רוצים אלגוריתם שיפתור את הבעיה **בתוחלת** זמן הקצרה ביותר.

ניתן לעשות זאת בתוחלת זמן :

א. $\Theta(\lg n)$

*ב. $\Theta(n)$

ג. $\Theta(n \lg n)$

ד. $\Theta(n^2)$

ה. אף אחד מהנ״ל

הסבר קצר (רשות) :

נכניס את כל האיברים לטבלת Hash בגודל n (פתרון התנגשויות על ידי רשימות מקושרות) בתוחלת זמן O(n). ערכים כפולים נשמור באותו איבר ברשימה על ידי מונה.

המונים הם שלמים בין 1 ל-n לכן אפשר למיין אותם במערך על ידי מיון מניה (CountSort) בזמן O(n). עכשיו בעזרת שני אינדקסים אפשר לרוץ על המערך מההתחלה ומהסוף, ולבדוק האם יש שני איברים במערך שסכומם 2004.

תעודת זהות :

מספר מחברת :

8. **(8)** נוסחת הנסיגה של זמן הריצה של התכנית Select (מציאת סדר סטטיסטי כלשהו במערך לא ממוין) היא :

$T(n) \le T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\frac{7n}{10} + c\right) + O(n)$ (c קבוע כלשהו).

נשנה את פעולת Select כך שבשלב הראשון החלוקה תהיה לשלישיות במקום לחמישיות. נוסחת הנסיגה המעודכנת תהיה :

א. $T(n) \le T\left(\left\lceil \frac{n}{3} \right\rceil\right) + T\left(\frac{n}{3} + c'\right) + O(n)$ ('c קבוע כלשהו).

*ב. $T(n) \le T\left(\left\lceil \frac{n}{3} \right\rceil\right) + T\left(\frac{2n}{3} + c'\right) + O(n)$ ('c קבוע כלשהו).

ג. $T(n) \le T\left(\left\lceil \frac{2n}{3} \right\rceil\right) + T\left(\frac{2n}{3} + c'\right) + O(n)$ ('c קבוע כלשהו).

ד. $T(n) \le T\left(\left\lceil \frac{n}{3} \right\rceil\right) + T\left(\frac{5n}{6} + c'\right) + O(n)$ ('c קבוע כלשהו).

ה. $T(n) \le T\left(\left\lceil \frac{n}{3} \right\rceil\right) + T\left(\frac{7n}{6} + c'\right) + O(n)$ ('c קבוע כלשהו).

ו. $T(n) \le T\left(\left\lceil \frac{n}{3} \right\rceil\right) + T\left(\frac{7n}{10} + c'\right) + O(n)$ ('c קבוע כלשהו).

הסבר קצר (רשות) : *מציאת חציון החציונים של השלשות:* $T\left(\left\lceil \frac{n}{3} \right\rceil\right)$*. מובטח שלפחות (כמעט) n/3 מהאיברים מסולקים לקראת השלב הבא.*

תעודת זהות:

מספר מחברת:

I.

א. (7)

רוצים אלגוריתם יעיל ככל הניתן לבעיה הבאה:

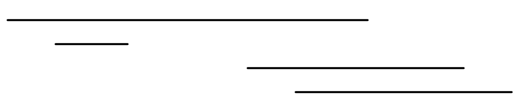
קלט: n קטעים על הישר (כלומר n זוגות של נקודות התחלה וסיום).

שאלה: האם יש שלושה קטעים בקלט שיש להם לפחות נקודת חיתוך אחת משותפת?

תאור האלגוריתם וניתוח זמן ריצה:

בונים מערך שמכיל את כל נקודות הקצה של כל הקטעים. מסמנים לכל נקודה אם היא הייתה נקודת התחלה או נקודת סיום. ראשית, ממיינים את המערך כך שהנקודות יהיו מסודרות משמאל לימין. אם יש "תיקו", כלומר נקודות שוות, אז שמים במערך את נקודות ההתחלה לפני נקודות הסיום. שנית, סורקים את המערך, ובכל שלב שומרים את מס' נקודות ההתחלה שעברנו פחות מס' נקודות הסיום שעברנו, פשוט ע"י הפחתה של 1 ממונה כשעוברים נק' סיום, והוספה של 1 בנק' התחלה. קל לראות שהמונה שווה בדיוק למס' הקטעים שהנק' הנוכחית נמצאת בהם, קל גם לראות שאם יש 3 קטעים שיש להם נקודה משותפת, אזי יש נקודה כזאת שהיא בפרט נק' קצה של קטע. לכן צריך להחזיר 'כן' אם ורק אם המונה מגיע באיזשהו שלב לערך שהוא לפחות 3. המיון לוקח זמן O(nlogn) והמעבר לוקח זמן לינארי, אז סה"כ זמן O(nlogn).

טעויות נפוצות: 1. היו פתרונות שמיינו את הקטעים לפי נק' התחלה, ואז בדקו אם ישנם שלושה קטעים סמוכים במערך שנחתכו באותה נקודה. זהו פתרון שגוי מכיוון שיכול להיות שיש שלשת קטעים שנחתכים באותה נק' בעוד שאין שלשה כזו של קטעים עם נק' התחלה סמוכות. למשל כמו בדוגמה הבאה:



2. היו גם פתרונות שבדקו רק אם יש שלשת קטעיםעוקבים שבה הראשון נחתך עם השני והשני נחתך עם השלישי, אך לא בדקו ששלושתם נחתכים בנקודה אחת.

תעודת זהות:

מספר מחברת:

ב. (10)

רוצים אלגוריתם יעיל ככל הניתן לבעיה הבאה:

קלט: n קטעים על הישר (כלומר n זוגות של נקודות התחלה וסיום).

שאלה: האם יש $\frac{n}{2}$ קטעים שיש להם לפחות נקודה אחת משותפת?

תאור האלגוריתם וניתוח זמן ריצה:

אותו פתרון מסעיף א׳ עובד גם כאן, רק שצריך לבדוק עבור הערך $\frac{n}{2}$ במקום הערך 3.

תעודת זהות :

מספר מחברת :

ג. (9)

האם ניתן למיין n מספרים $a_1,\ldots,a_n$ מהתחום [0,2] בזמן .W.C $O(n)$ אם

לכל $i\neq j$ מתקיים $|a_i-a_j|>1/n^2$?

תשובה : *כן*

הוכחה :

לכל מספר ai *"נצמיד" מספר* ni *כך שיתקיים* ni/2n2<= ai<= (ni+1)/2n2.
מהמנתון נקבל כי כל המספרים ni*, הינם שונים, וברור כי מיון המספרים* ni *נותן מיון של המספרים* ai*. את המספרים* ni *ניתן למיין ב* O(n) *כפי שראינו בתרגול.*

תעודת זהות :

מספר מחברת :

ד. **(7)**

תנו חסם תחתון במודל ההשוואות לבעיה מסעיף ג'.

החסם : nlogn

הוכחה :

במיון בהשוואות לא משנה מה הם ערכי האיברים, כי בהוכחת החסם התחתון משתמשים רק בעובדה שמותר רק להשוות זוגות איברים ועלות השוואה של שני מספרים היא פעולה אחת. בשאלה הנידונה אפשר לראות שמספר האברים השונים זה מזה הינו N ולכן מספר הסדרים האפשריים, כמו בהוכחה המקורית הינו N!. לכן ההוכחה המקורית תופסת גם כאן.

סוף

מספר קורס: 0368.2158

סמסטר א', מועד א', תשס"ד

26/2/2004

מבחן במבני נתונים

פרופ' חנוך לוי, עודד שוורץ ואסף שפירא

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים. מותר להביא דף עזר אחד.

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף בטופס זה.
2. יש לענות על כל השאלות.
3. הסברים לשאלות האמריקאיות – רשות.
4. **תשובות לשאלות הפתוחות והסברים לשאלות האמריקאיות יש לכתוב <u>על טופס המבחן בלבד</u>.** המחברות הן לטיוטה בלבד ולא תיבדקנה.
5. **תשובות סופיות לשאלות האמריקאיות יש לכתוב אך ורק <u>על הטופס המצורף</u>**. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. **<u>ניתן וצריך</u>** להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. במבחן 100 נקודות. ניקוד פרטני ליד השאלות.
8. **<u>הערה חשובה לשאלות הפתוחות:</u>** כתבו תשובות קצרות ומדויקות. תשובות לא ממוקדות/ ארוכות/ מסובכות/ מסורבלות יקבלו ניקוד חלקי **גם אם הן נכונות**. כמו כן, **אין** לכתוב אלגוריתמים בפסאודו-קוד אלא להסביר (באופן משכנע) את הרעיון הכללי.

בהצלחה!!

1. **(4)**

עץ *טרינארי* הוא עץ שבו לכל צומת לכל היותר שלושה ילדים.

נתון **<u>עץ טרינארי מלא ומאוזן</u>**, המכיל **N** צמתים.

העומק של צומת הינו מספר הקשתות בין השורש ובין הצומת (עומק השורש הינו 0).

הגובה של צומת הינו מספר הקשתות בין הצומת ובין העלה הקרוב ביותר (גובה עלה הינו 0).

יהי **D סכום העומקים של כל הצמתים (כולל עלים)** אזי

א. $D = \Theta(N)$

ב. $D = \Theta(N \log N)$

ג. $D = \Theta(N^2)$

ד. $D = \Theta(N^2 \log N)$

ה. אף אחת מהתשובות הנ"ל אינה נכונה.

הסבר קצר (רשות):
ב.
עומק כל הצמתים הוא O(lg n). לכן הסכום הוא O(n lg n).
מצד שני, עומק כל העלים הוא Ω(lg n) ויש Ω(n) עלים לכן
הסכום הוא גם Ω(n lg n)

2. **(4)**

אותם נתונים כמו בשאלה הקודמת.

יהי **H סכום הגבהים של כל הצמתים (כולל עלים)** אזי

א. $H = \Theta(N)$

ב. $H = \Theta(N \log N)$

ג. $H = \Theta(N^2)$

ד. $H = \Theta(N^2 \log N)$

ה. אף אחת מהתשובות הנ"ל אינה נכונה.

הסבר קצר (רשות):
א.
החישוב בדומה לחישוב זמן הריצה של Build-Heap. מתקבלת
סדרה הנדסית מהצורה $\Theta(n) \times \Sigma_i (i/3^i)$.

3. (7)

$$T(n)=T(\frac{n}{4})+T(\frac{3n}{4})+1$$

א. $T(n)=\Theta(n)$

ב. $T(n)=\Theta(n^{\log_3 4})$

ג. $T(n)=\Theta(n\log n)$

ד. $T(n)=\Theta(n^{\frac{3}{4}})$

ה. $T(n)=\Theta(\lg^2 n)$

ו. $T(n)=\Theta(\lg^{\frac{3}{4}} n)$

הסבר קצר (רשות):
א.
ראינו בכיתה.

4. (7)

נתונה טבלת Hash פתוחה בגודל B, שבה התנגשויות בכל תא נפתרות על ידי רשימה מקושרת.

הטבלה תומכת בפעולות Insert, Delete, Find.

עבור סדרת פעולות באורך N, המורכבת ממספר שווה של כל אחת מהפעולות הנ"ל (N/3 מכל סוג), ומתחילה עם מבנה ריק, זמן Amortized לפעולה בודדת במבנה הוא:

(תזכורת: זמן Amortized לפעולה הוא זמן .W.C של סידרה של N פעולות, חלקי N).

א. $\Theta(1)$

ב. $\Theta(B)$

ג. $\Theta(N)$

ד. $\Theta(B^2)$

ה. $\Theta(N^2)$

ו. אף אחת מהנ"ל.

הסבר קצר (רשות):
ג. כל פעולה במבנה לוקחת O(N) w.c. ולכן זהו חסם על זמן Amortized. סידרה שמתחילה N/3 פעולות Insert שבה כל האיברים נופלים לאותו תא, וכל פעולות החיפוש כושלות אחרי סריקת הרשימה הארוכה, תיקח $\Theta(N^2)$.

5. **(7)**

אם $\log(f(n)) = \Theta(\log(g(n)))$ אז בהכרח:

א. $f(n) = \Theta(g(n+1))$

ב. $f(n) = \Theta(g(\frac{n}{2}))$

ג. $f(n) = \Theta(g(\sqrt{n}))$

ד. א + ב

ה. א + ב + ג

ו. אף אחת מהנ"ל

הסבר קצר (רשות):
ו.
דוגמה נגדית:
$f(n) = n!$, $g(n) = n^n$

6. **(7)**

נתון עץ חיפוש בינארי התומך בפעולות סדר סטטיסטי (Order Statistic). גובה העץ בקשתות הוא h ויש בו n איברים. מבצעים על העץ סריקת LDR (In-order), אלא שבמקום להדפיס את המפתחות, מדפיסים את ערך השדה Size.

הערך המקסימלי יודפס **בהכרח** במקום:

א. הראשון

ב. האחרון

ג. אחד מ-h המקומות הראשונים.

ד. אחד מ-h המקומות האחרונים.

ה. n/2

ו. אף אחד מהנ"ל.

הסבר קצר (רשות):
ו.
הערך Size המקסימאלי שייך לשורש. השורש יכול להיות כל סדר סטטיסטי, ולכן יכול להופיע בכל מקום בהדפסת Inorder.

7. **(7)**

נתון מערך ממוין שבו כל איבר מופיע $\sqrt{n}$ פעמים. ניתן לבצע חיפוש איבר במערך זה בזמן W.C.

א. $\Theta(\lg n)$

ב. $\Theta\left(\sqrt{\lg n}\right)$

ג. $\Theta\left(\sqrt{n}\right)$

ד. $\Theta\left(\frac{\sqrt{n}}{\lg n}\right)$

ה. $\Theta(\lg\lg n)$

ו. $\Theta\left(\lg^2 n\right)$

הסבר קצר (רשות):
א. אם נבצע חיפוש בינארי רגיל (ונעצור אותו כשהדילוגים קטנים מ-$\sqrt{n}$ זמן הריצה יהיה (W.C.) $\Theta(\lg\sqrt{n}) = \Theta(\lg n)$

8. **(7)**

נשנה את אלגוריתם QuickSort באופן הבא: בחירת איבר הציר (Pivot) תבוצע על ידי בחירת הסדר הסטטיסטי ה- $\frac{n}{4}$, בעזרת פונקצית Select. זמן ריצת האלגוריתם החדש יהיה:

א. $W.C. - \Theta(n\lg n), Expectet - \Theta(n)$

ב. $W.C. - \Theta\left(n^2\right), Expectet - \Theta\left(n^2\right)$

ג. $W.C. - \Theta\left(n^2\right), Expectet - \Theta(n\lg n)$

ד. $W.C. - \Theta(n\lg n), Expectet - \Theta(n\lg n)$

ה. $W.C. - \Theta\left(n^{\frac{1}{4}}\lg n\right), Expectet - \Theta\left(n^{\frac{1}{4}}\lg n\right)$

ו. $W.C. - \Theta\left(n^2\lg n\right), Expectet - \Theta\left(n^2\lg n\right)$

הסבר קצר (רשות):
ד.
זמן הריצה יהיה זהה למקרה הגרוע ולמקרה הממוצע – למעשה לא יושפע מהקלט או מהגרלות.
נוסחת הנסיגה: $T(n) = T(3n/4) + T(n/4) + \Theta(n)$ (עד כדי עיגול לשלמים).

9. **(7)** נוסחת הנסיגה של זמן הריצה של התכנית Select (מציאת סדר סטטיסטי כלשהו במערך לא ממוין) היא :

$$T\left(n\right)\le T\left(\left\lceil \frac{n}{5}\right\rceil\right)+T\left(\frac{7n}{10}+c\right)+O\left(n\right)$$ (c קבוע כלשהו).

נשנה את פעולת Select כך שבשלב הראשון החלוקה תהיה לשביעיות במקום לחמישיות. נוסחת הנסיגה המעודכנת תהיה :

א. $T\left(n\right)\le T\left(\left\lceil \frac{n}{7}\right\rceil\right)+T\left(\frac{7n}{10}+c'\right)+O\left(n\right)$ (c' קבוע כלשהו).

ב. $T\left(n\right)\le T\left(\left\lceil \frac{n}{7}\right\rceil\right)+T\left(\frac{n}{2}+c'\right)+O\left(n\right)$ (c' קבוע כלשהו).

ג. $T\left(n\right)\le T\left(\left\lceil \frac{n}{7}\right\rceil\right)+T\left(\frac{n}{10}+c'\right)+O\left(n\right)$ (c' קבוע כלשהו).

ד. $T\left(n\right)\le T\left(\left\lceil \frac{n}{7}\right\rceil\right)+T\left(\frac{2n}{7}+c'\right)+O\left(n\right)$ (c' קבוע כלשהו).

ה. $T\left(n\right)\le T\left(\left\lceil \frac{n}{7}\right\rceil\right)+T\left(\frac{5n}{7}+c'\right)+O\left(n\right)$ (c' קבוע כלשהו).

ו. $T\left(n\right)\le T\left(\left\lceil \frac{n}{5}\right\rceil\right)+T\left(\frac{7n}{10}+c'\right)+O\left(n\right)$ (c' קבוע כלשהו).

הסבר קצר (רשות): ה. מציאת חציון החציונים היא הפעלה רקורסיבית על מערך בגודל בערך n/7. יש בערך 4n/14 איברים בהכרח קטנים ממנו וכנ"ל גדולים (חצי מארבע שורותבגודל n/7 כל אחת) לכן הרקורסיה השניה פועלת על מערכ בגודל בערכ 5n/7

10. **(3)** אותם נתונים כמו בשאלה הקודמת. מה זמן הריצה .W.C של התכנית החדשה?

א. $\Theta\left(\lg n\right)$

ב. $\Theta\left(n\right)$

ג. $\Theta\left(n\lg n\right)$

ד. $\Theta\left(n^2\right)$

ה. אף אחד מהנ"ל

הסבר קצר (רשות): ב. הוכחה באינדוקציה.

I **(15)** רוצים להוסיף לעץ 2-3 רגיל (התומך בפעולות insert, delete, find) את הפעולה הבאה:
Change-Key(p,new-val). הפעולה מקבלת מצביע p לאיבר (עלה) בעץ, וצריכה **לעדכן** את הערך השמור באיבר עליו מצביע p לערך new-val. שימו לב כי הפעולה מקבלת **מצביע** לאיבר עליו יש לבצע את הפעולה. בסיום הפעולה, העץ צריך להיות עץ 2-3 חוקי.
הוכיחו חסמים עליונים ותחתונים הדוקים (כלומר זהים) לביצוע הפעולה הנ"ל במודל ההשוואות.

(5) חסם עליון: $O(\log n)$

הוכחה: ניתן למחוק את p.key ב $O(\log n)$ ולהכניס את new-val ב $O(\log n)$. סה"כ $O(\log n)$.

(10) חסם תחתון: $\Omega(\log n)$

הוכחה: נניח כי ניתן לבצע את הפעולה ב $o(\log n)$. בהינתן n מספרים לא ממוינים נמיינם באופן הבא:

א. נבנה עץ 2-3 חוקי שכל עליו מכילים את הערך 0. כפי שראינו ניתן לעשות זאת ב $O(n)$.

ב. נקצה מערך נוסף B אשר יחזיק n מצביעים לעליו של העץ. גם את מערך זה ניתן לבנות ב $O(n)$ בזמן בניית העץ.

ג. נעבור על מערך הקלט ועבור האיבר ונבצע ChangeKey(B[i],A[i]). כך נקבל עץ 2-3 חוקי המכיל את איברי המערך וזאת בזמן o(nlong).

ד. נוציא מהעץ רשימה מקושרת המכילה את עליו ב $O(n)$.

סה"כ קיבלנו מיון ב $o(n\log n)$ בסתירה לחסם התחתון של מודל ההשוואות.

II **(15)**

(א) (5) נתונים n מפתחות שונים זה מזה וממוינים. תארו **מבנה נתונים** __**יעיל**__ המתחזק מפתחות אלו (הניחו שהמבנה כבר בנוי, כלומר לא חשובה עלות בנייתו) והתומך בפעולת find_delete(key) המקבלת מפתח והמחזירה false אם המפתח אינו במבנה. אם הוא במבנה, הפעולה מחזירה true ומוציאה אותו מהמבנה. עלות נדרשת : O(log n) לפעולת find_delete(key).

תאור המבנה :

עץ חיפוש מאוזן.

סבוכיות הפעולה – הסבירו :

מציאת איבר בעץ, וכן מחיקתו עולות O(logn).

(ב) (10) נגדיר פרמוטציה מסדר (n,m) באופן הבא (כפי שהוגדר בתרגול):

א. מסדרים את המספרים 1,2,3,…,n במעגל, ומצביעים על המספר 1.

ב. מבצעים m צעדים לאורך המספרים **שנותרו** על המעגל.

ג. מדפיסים את המספר אליו הגענו, ומוחקים אותו מהמעגל. אם נותרו מספרים חוזרים לב', ולא עוצרים. מספר שנמחק, **אינו** נספר במניין הצעדים בשלב ב'.

לדוגמא: הפרמוטציה מסדר (7,3) היא 4,7,3,1,6,2,5 (ודאו כי אתם מבינים מדוע!)

תארו אלגוריתם **<u>יעיל</u>** המקבל מספר n, ומחזיר את הפרמוטציה מסדר (n,n/2).

(שימו לב, שהרעיון לא בהכרח דומה לזה שנלמד בכיתה עבור הפרמוטציה (n,17)).

זמן ריצת האלגוריתם: O(nlogn)

תיאור האלגוריתם והוכחת נכונות:

נחזיק את המספרים 1 עד n בעץ חיפוש בינארי עם סדר סטטיסטי. ז"א כל קודקוד בעץ ישמור בנוסף את מספר הקודקודים שמתחתיו (+1) בשדה size, כמו שראינו בכיתה.
נתחיל כשפוינטר מצביע על האיבר 1.

1) כשאנחנו באיבר שאינדקסו I, נקפוץ לאיבר שאינדקס שלו הוא (I+n/2) mod k בעזרת OS_Select כאשר k הוא מספר האיברים שנותרו בעץ (כלומר size של השורש).

2) נדפיס את האיבר שהגענו אליו, ונמחק אותו.

3) נבצע את 1 ו- 2 עד שלא יותרו איברים בעץ.

לאלג' יש n איטרציות. כל איטרציה עולה O(logn), וסה"כ O(nlogn).

טעויות נפוצות:

אלג' שזמנם O(n^2) – ניתנו 2 נק'

אלג' שלא בצעו את הנדרש לא הדפיסו את הפרמוטציה –
לא קיבלו הוב'

ניתוח זמן ריצה:

III: **(10)**

א. **(10)** תאר אלגוריתם הפותר את הבעיה הבאה: בהינתן קבוצה S של n מספרים, האלגוריתם בונה בזמן **W.C. O(n)**, מבנה נתונים התומך בפעולה find בלבד, באופן ש: זמן ה W.C של find הוא **O(n)**, אולם על **לפחות n/log n** מהאיברים, זמן הריצה של find הוא **O(log n)**. שימו לב כי המבנה אינו דינאמי, כלומר אינו צריך לתמוך ב insert ו delete.

תיאור האלגוריתם והוכחת נכונות:

לוקחים n/log n איברים ומכניסים לעץ חפוש מאוזן (לדוגמא 2-3) או למערך ממוין. עלות כוללת היא O(n). שאר האיברים מוכנסים לרשימה.

בעת ביצוע find מחפשים קודם בעץ ואחר כך ברשימה. עלות חפוש על כל איבר בעץ היא O(log n). שאר האיברים – O(n).

הוכחת זמן ריצה:

סוף

מספר מחברת:

מספר קורס: 0368.2158

סמסטר א'+ב', מועד ב', תשס"ב

8/10/2003

מבחן במבני נתונים

פרופ' חנוך לוי, פרופ' יוסי מטיאס, עודד שוורץ, אסף שפירא

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים.

מותר להשתמש בדף עזר אחד

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף בטופס זה.
2. יש לענות על כל השאלות.
3. **תשובות לשאלות הפתוחות יש לכתוב <u>על טופס המבחן בלבד</u>.** המחברות הן לטיוטה בלבד ולא תיבדקנה.
4. **תשובות לשאלות האמריקאיות יש לכתוב אך ורק <u>על הטופס המצורף</u>.** בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
5. **הערה חשובה לגבי השאלות הפתוחות**: תשובות ארוכות/מסורבלות/מסובכות/לא ברורות לא יזכו לניקוד מלא <u>גם אם הן נכונות</u>. ענו תשובות קצרות ומדויקות. ניתן להשתמש כ"קופסא שחורה" בכל אלגוריתם/משפט שנלמד בכיתה.
6. שאלות אמריקאיות 8 נקודות. שאלות פתוחות 20 נקודות. אפשר לצבור עד 100 נקודות.

בהצלחה!!

שאלה	ציון
9	
10 א	
10 ב	
סה"כ	

.1

$$T(n) = T(\frac{n}{7}) + T(\frac{n}{2}) + 6n$$
$$T(n) = ?$$

א. $\Theta(n^2)$

ב. $\Theta(n \lg n)$

ג. $\Theta(n)$

ד. $\Theta(\lg n \lg \lg n)$

ה. $\Theta(\lg n)$

ו. אף אחד מהנ״ל.

.2

$$f(n) = 2^{\lg\lg n}, g(n) = 4^{\sqrt{\lg\lg n}}$$

מה מהבאים נכון ?

א. f = Θ(g)

ב. f = o(g) (o קטן).

ג. g = o(f) (o קטן).

ד. אף אחד מהנ״ל

3. נתון מבנה A שתומך בפעולות Insert, Delete, Find. אם נתון שזמן Amortized לפעולה הוא $\Theta(1)$, אז יתכן שפעולת Insert לוקחת ב-.W.C:

א. $\Theta(\lg\lg n)$

ב. $\Theta(\lg n)$

ג. $\Theta(n)$

ד. א+ב

ה. ב+ג

ו. א+ב+ג

4. נתון מערך בגודל n^3 שבו יש n^2 ערכים שונים. אלגוריתם Quick-Sort, ירוץ עליו בזמן W.C. :

א. $\Theta(n^2 \lg n)$

ב. $\Theta(n^3)$

ג. $\Theta(n^3 \lg n)$

ד. $\Theta(n^3 \lg^3 n)$

ה. $\Theta(n^4)$

ו. $\Theta(n^6)$

5. נתונה ערימת מקסימום המכילה n איברים. מובטח שהערימה חוקית מבחינת מבנה, אך יתכן שיש הפרה אחת (לכל היותר) של חוקיות הערכים בין אבא לבן. רוצים להכריע האם יש הפרה כזו. האלגוריתם היעיל ביותר לבדיקה הנ"ל ירוץ בזמן WC :

א. $\Theta(n^2)$

ב. $\Theta(n\lg n)$

ג. $\Theta(n)$

ד. $\Theta(\lg n)$

ה. אף אחד מהנ"ל.

6. נתון מבנה נתונים X המכיל n איברים. רוצים למצוא את האיבר הקטן ביותר שגדול מהאיבר המינימלי (כלומר השני בגודלו). איזה מבין המבנים הבאים יאפשר את הנ"ל בזמן .W.C המהיר ביותר ? (הכוונה לשימוש במבנה כמו שנלמד בכיתה, ללא שום שינוי)

א. ערימת מקסימום

ב. Perfect Hash

ג. עץ חיפוש מאוזן התומך בפעולות Order-Statistic

ד. תור

ה. מערך ממוין

ו. יש יותר מתשובה אחת נכונה.

7. נתון עץ חיפוש בינרי (לאו דווקא מאוזן) בגודל n ובגובה h (בקשתות). לכל צומת שדה נוסף המכיל את המספר $\frac{1}{2^d}$ כאשר d הוא המרחק (בקשתות) של הצומת משורש הערימה. אם נסכום את כל השדות הנ״ל של **כל צמתי העץ** נקבל מספר W. מהם החסמים (תחתון ועליון) **ההדוקים ביותר** עבור W ?

א. $\frac{1}{2}-\frac{1}{2^h} \le W \le \frac{1}{2}$

ב. $0 \le W \le n$

ג. $0 \le W \le 1$

ד. $1-\frac{1}{2^h} \le W \le n\lg n$

ה. $1-\frac{1}{2^h} \le W \le 1$

ו. $1-\frac{1}{2^h} \le W \le n$

8.

מספר מחברת:

9. רוצים אלגוריתם יעיל ככל האפשר (WC) לבעיה הבאה:

קלט: מערך A בגודל n ומספר k.

פלט: k האיברים הקרובים ביותר לחציון המערך שמרחקם שההפרש בינם לחציון לפחות 10.

מותר להניח ש-n אי זוגי ושתמיד יש פתרון אחד בלבד.

לדוגמה: עבור הקלט A = 24, 23, 35, 32, 47, 10, 11 , k=3

הפלט יהיה 35,10,11

הציעו אלגוריתם שפותר את הבעיה בזמן WC הטוב ביותר.

זמן ריצה:

תאור האלגוריתם:

הוכחת זמן ריצה:

מספר מחברת:

10. א.

ערימת-מקסימום תיקרא *"חזקה"* אם בכל שכבת צמתים, כל הצמתים קטנים מכל הצמתים בשכבה שמעליה.

תארו אלגוריתם בעל סיבוכיות ה **W.C.** הטובה ביותר לבניית ערימת-מקסימום חזקה.

זמן ריצה:

תיאור האלגוריתם:

הוכחת זמן ריצה:

מספר מחברת:

11. ב.

ערמת-מקסימום תיקרא *"חזקה מאד"* אם בכל שכבת צמתים, כל הצמתים קטנים מכל הצמתים בשכבה שמעליה, וכן כל **עלה** בערימה גדול מהעלה משמאלו.

תנו חסם תחתון הדוק ביותר לזמן ריצת WC של אלגוריתם לבניית ערימה חזקה מאוד (בהנחת מודל ההשוואות).

החסם:

הוכחה:

מספר תעודת זהות: ______

מספר מחברת: ______

מספר קורס: 0368.2158

סמסטר ב׳, מועד א׳, תשס״ב

6/7/2003

מבחן במבני נתונים

פרופ׳ יוסי מטיאס, אסף שפירא, עודד שוורץ

משך המבחן שלוש וחצי שעות (אין הארכה)
הבחינה עם חומר סגור, ללא מחשבים/מחשבונים.
מותר להשתמש בדף עזר אחד

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף בטופס זה.
2. יש לענות על כל השאלות.
3. **תשובות לשאלות הפתוחות יש לכתוב <u>על טופס המבחן בלבד</u>.** המחברות הן לטיוטה בלבד ולא תיבדקנה.
4. **תשובות לשאלות האמריקאיות יש לכתוב אך ורק <u>על הטופס המצורף</u>**. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
5. **הערה חשובה לגבי השאלות הפתוחות**: תשובות ארוכות/מסורבלות/מסובכות/לא ברורות לא יזכו לניקוד מלא <u>גם אם הן נכונות</u>. ענו תשובות קצרות ומדויקות. ניתן להשתמש כ״קופסא שחורה״ בכל אלגוריתם/משפט שנלמד בכיתה.
6. שאלות אמריקאיות 8 נקודות. שאלות פתוחות 20 נקודות. אפשר לצבור עד 100 נקודות.

בהצלחה!!

שאלה	ציון
9	
10 א	
10 ב	
סה״כ	

1. נניח כי עומד לרשותכם אלגוריתם המסוגל לממש את פעולת Heapify (שהוגדרה בכיתה) על ערימה בזמן O(lglg(n)). מה יהיה זמן הריצה של אלגוריתם BuildHeap (כפי שהוגדר בכיתה), אם במקום להשתמש באלגוריתם Heapify הרגיל כקופסא שחורה, תשתמשו באלגוריתם Heapify החדש?

א. $\Theta(n\lg n)$

ב. $\Theta(n\lg\lg n)$

ג. $\Theta(n\lg n/\lg\lg n)$

ד. $\Theta(n\lg\lg n/\lg n)$

**ה. $\Theta(n)$

2. בכיתה ראינו אלגוריתם חיפוש בינרי במערך ממוין.

נשנה אותו באופן הבא:

בכל שלב, במקום לבדוק רק את הערך באינדקס $\frac{n}{2}$, נבדוק את הערך בכל אחד מהאינדקסים $\frac{n}{k},2\frac{n}{k},3\frac{n}{k},...,(k-1)\frac{n}{k}$.

אם באחד מהם הערך שווה לערך שמחפשים – נעצור ונחזיר את האינדקס שלו.

אחרת – נפעיל את האלגוריתם באופן רקורסיבי רק על הקטע המתאים.

זמן הריצה של אלגוריתם זה יהיה:

א. $\Theta(n)$

ב. $\Theta(n\lg k)$

**ג. $\Theta(k\lg_k n)$

ד. $\Theta(\lg_k n)$

ה. $\Theta(\frac{\lg n}{k})$

3.
$$T(n) = T(n-1) + \lg n$$
$$T(n) = ?$$

א. $\Theta(n^2)$

**ב. $\Theta(n \lg n)$

ג. $\Theta(n)$

ד. $\Theta(\lg n \lg\lg n)$

ה. $\Theta(\lg n)$

ו. אף אחד מהנ״ל.

4.
$$f(n) = 2^{\sqrt{\lg\lg n}}, g(n) = (\lg\lg n)^2 \sqrt{n}$$

מה מהבאים נכון ?

א. f = Θ(g)

ב. f = o(g) (o קטן).****

ג. g = o(f) (o קטן).

ד. אף אחד מהנ״ל

5. נתונה ערימת מקסימום (Heap), שבה לכל צומת שדה נוסף המכיל את המספר $\frac{1}{2^d}$ כאשר d הוא המרחק (בקשתות) של הצומת משורש הערימה. אם נסכום את כל השדות הנ״ל של **עלי הערימה בלבד** נקבל מספר W. מהם החסמים (תחתון ועליון) **ההדוקים ביותר** עבור W ?

א. $0 \le W \le n$

ב. $\frac{1}{2} \le W \le 1$

ג. $\frac{1}{2}-\frac{1}{2^h} \le W \le 1$ כאשר h הוא גובה הערימה (בקשתות)

ד. $\frac{1}{2}-\frac{1}{2^h} \le W \le \frac{1}{2}$ כאשר h הוא גובה הערימה (בקשתות)

**ה. $1-\frac{1}{2^h} \le W \le 1$ כאשר h הוא גובה הערימה (בקשתות)

ו. $\frac{1}{2} \le W \le \frac{1}{2}+\frac{1}{2^h}$ כאשר h הוא גובה הערימה (בקשתות)

6. רוצים להפוך עץ חיפוש בינרי לעץ חיפוש אדום שחור. בהנחת מודל ההשוואות ניתן לעשות זאת בזמן (הטוב ביותר האפשרי):

**א. $\Theta(n)$ W.C., $\Theta(n)$ ממוצע.

ב. $\Theta(n)$ W.C., $\Theta(n\lg n)$ ממוצע.

ג. $\Theta(n\lg n)$ W.C., $\Theta(n)$ ממוצע.

ד. $\Theta(n\lg n)$ W.C., $\Theta(n\lg n)$ ממוצע.

ה. אף אחד מהנ״ל

ו. א+ב+ג

7. נתונים שני מערכים **ממוינים** בגודל n. אף ערך לא חוזר על עצמו. רוצים למצוא את החציון המשותף שלהם (כלומר איבר שיש בדיוק n איברים שקטנים ממנו בששני המערכים יחד).

ניתן לעשות זאת בזמן W.C. (הטוב ביותר):

א. $\Theta(1)$

**ב. $\Theta(\lg n)$

ג. $\Theta(n)$

ד. $\Theta(n\lg n)$

ה. $\Theta(n^2)$

8. נתון עץ AVLבגודל n. מבצעים פעולת רוטציה על שורש העץ. נזכיר שעומק של צומת הוא מרחקו מהשורש.

כמה צמתים **לא** שינו את עמקם?

א. 0

ב. $\Theta(1)$

ג. $\Theta(\lg n)$

**ד. $\Theta(n)$

ה. אף אחד מהנ״ל לא בהכרח נכון (תלוי בעץ עצמו).

9. נתון מערך A המכיל n מספרים שלמים. בהינתן מספר x המופיע ב A, נגדיר את הריבוי של x במערך A בתור מספר המופעים של x ב-A. רוצים לייצר מערך <u>ממוין</u> B בגודל n, שיכיל את הריבוי של כל אחד מהמספרים המופיעים ב A.

הציעו אלגוריתם שפותר את הבעיה **בתוחלת** זמן הטובה ביותר.

תוחלת זמן ריצה:O(n)

תאור האלגוריתם:

נכניס את איברי הקבוצה לטבלת Hash בגודל $\Theta(n)$. ערכים חוזרים יוחלקו על ידי מונה (כלומר בפעולת ההכנסה, נעבור על כל הרשימה בתא אליו הגענו, ורק אם הערך לא נמצא נוסיף איבר חדש לרשימה, עם מונה מאותחל לאחד). בסוף הכנסת כל האיברים, נעבור על הטבלא ונחלץ את כל הריבויים. כל הערכים הנ"ל הם שלמים בתחום 1 עד n לכן ניתן למינם בעזרת CountSort.

הוכחת זמן ריצה :

הכנסה לטבלה – סה"כ צפוי (ממוצע) O(n)

חילוץ ריבויים מהטבלה O(n)

CountSort O(n)

סה"כ צפוי O(n)

מספר מחברת:

10. בהינתן שני מערכים A ו B, כל אחד בגודל n, מעוניינים לייצר מערך שלישי C בגודל n, בעל התכונה הבאה: הערך הרשום ב C[i] שווה מספר איברי A הקטנים מ B[i].

לשאלה זו שני סעיפים!

א. תארו אלגוריתם בעל סיבוכיות ה **W.C.** הטובה ביותר לפתרון הבעיה.

זמן ריצה: O(n lg n)

תיאור האלגוריתם:

נמיין את A בעזרת HeapSort.

עבור כל איבר ב-B נמצא כמה איברים קטנים ממנו בעזרת חיפוש בינארי ב O(lg n).

יש איברים המופיעים יותר מפעם אחת, חיפוש בA חשוב מאוד: אם ב- בינארי רגיל לא יעבוד. לכן בכל דגימה של ערך, אם הערך הנדגם שווה לערך שמחפשים האלגוריתם לא יעצור אלא יבצע קריאה רקורסיבית על החצי השמאלי (שמכיל את האיברים הקטנים יותר).

הוכחת זמן ריצה:

מיון: $\Theta(n \lg n)$

n *חיפושים בינארים* $\Theta(n \lg n)$

סה"כ $\Theta(n \lg n)$

ב. תנו חסם תחתון טוב ככל האפשר לאלגוריתם הפותר את הבעיה במודל ההשוואות :

הוכחת נכונות :

נראה שבעזרת אלגוריתם לפתרון הבעיה הנ"ל שפועל במודל ההשוואות בזמן f(n) *ניתן למיין מערך המכיל את הערכים 1 עד* n *בזמן* O(f(n)+n) *ולכן בהכרח* $f(n)=\Omega(n \lg n)$.

בהינתן מערך A *המכיל את* n *המספרים הראשונים, נעביר לאלגוריתם שני עותקים של* A *(בתור* A *ו-*B).
*האלגוריתם פועל ב-*f(n) *ומחזיר מערך* C *המכיל למעשה את הסדר הסטטיסטי של כל אחד מאברי* A *ולכן בזמן נוסף* O(n) *ניתן למיין את* A.

מספר מחברת:

מספר קורס: 0368.2158

סמסטר א׳, מועד א׳, תשס״ג

13/2/2003

מבחן במבני נתונים

פרופ׳ חנוך לוי, אסף שפירא ועודד שוורץ

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים.

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף בטופס זה.
2. יש לענות על כל השאלות.
3. יש לתת הסברים לכל השאלות (**כולל האמריקאיות**).
4. **תשובות לשאלות הפתוחות והסברים לשאלות האמריקאיות יש לכתוב <u>על טופס המבחן בלבד</u>.** המחברות הן לטיוטה בלבד ולא תיבדקנה.
5. **תשובות סופיות לשאלות האמריקאיות יש לכתוב אך ורק <u>על הטופס המצורף</u>**. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
6. ניתן להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. במבחן 100 נקודות. ניקוד פרטני ליד השאלות.

בהצלחה!!

1. (8)

$$T(n) = T(\frac{n}{4}) + T(\frac{3n}{4}) + 1$$

רמז: אפשר להשתמש בעץ רקורסיה.

VVא. $T(n) = \Theta(n)$

ב. $T(n) = \Theta(n^{\log_3 4})$

ג. $T(n) = \Theta(n \log n)$

ד. $T(n) = \Theta(n^{\frac{3}{4}})$

הסבר קצר:

בעזרת שיטת עץ הרקורסיה, או הוכחה באינדוקציה.

2. (4) אם $f(n) = \Theta(g(n))$ (ושתיהן פונקציות מונוטוניות עולות מהטבעיים לטבעיים) אז בהכרח:

$$\log(f(n)) = \Theta(\log(g(n)))$$

VVא. נכון

ב. לא נכון

הסבר קצר:

קיים קבוע c וקיים n' כך שלכל n>n' מתקיים:
f(n)<c g(n) . log היא פונקציה מונוטונית עולה, לכן
log(f(n))<log (c) + log (g(n)), לכן עבור כל קבוע d>log(c)+1
מתקיים:
log(f(n))<d log(g(n)) לכן log(f(n))=O(log(g(n))
באופן דומה מראים את החסם בכיוון השני.

3. **(4)** אם $f(n)=\Theta(g(n))$ (ושתיהן פונקציות מונוטוניות עולות מהטבעיים לטבעיים) אז בהכרח:

$$2^{f(n)}=\Theta(2^{g(n)})$$

א. נכון

VVב. לא נכון

הסבר קצר:
דוגמה נגדית: f(n) = 2 log(n), g(n) = 3 log(n)

4. **(8)** בסריקת DLR (PreOrder) של עץ חיפוש בגובה h יופיע האיבר המינימלי במקום:

א. הראשון

ב. האחרון

VVג. אחד מ-h המקומות הראשונים.

ד. אחד מ-h המקומות האחרונים.

ה. אף אחד מהנ״ל.

הסבר קצר:
האיברים הראשונים שמודפסים הם האיברים על המסלול מהשורש שמאלה. יש לכל היותר h איברים במסלול זה, והאיבר המינימלי נמצא בסופו.

5. **(8)** נתון מבנה נתונים התומך בפעולות Insert, ExtractMax (לא בהכרח במימוש ערימה).
בהנחת מודל ההשוואות, תנו חסם תחתון (הגבוה ביותר) לזמן Amortize של פעולה במבנה.
(תזכורת – בחישוב זמן Amortize הסידרה יכולה להכיל כל רצף של פעולות, והננו מעוניינים בסדרה המוליכה לזמן הגרוע ביותר).

א. $\Omega(1)$

ב. $\Omega(\log\log n)$

VVג. $\Omega(\log n)$

ד. $\Omega(n)$

ה. $\Omega(n\log n)$

ו. $\Omega(\frac{n}{\log n})$

הסבר קצר:
על ידי n פעולות insert ואז n פעולות ExtractMax נקבל מיון של n איברים.
לכן זמן ריצה של סידרה כזו, בהנחת מודל ההשוואות חייב להיות $\Omega(n\log n)$, לכן זמן Amortize של פעולה במבנה הוא $\Omega(\log n)$.

6. **(8)** נתון מערך בגודל n. רוצים אלגוריתם המוצא שלשה של איברים במערך: x,y,z כך ש:
x + y = z. אם אין שלשה כזו, האלגוריתם מחזיר 'לא'.
האלגוריתם המהיר ביותר (ממוצע) לפתרון הבעיה פועל בזמן **ממוצע:**

VVא. $O(n^2)$

ב. $O(n^2\log n)$

ג. $O(n^3)$

ד. $O(n^3\log n)$

ה. $O(n^2\log^2 n)$

הסבר קצר: הסבר קצר: נכניס את כל הסכומים המתקבלים מזוגות של מספרים לטבלתHash בזמן ממוצע $O(n^2)$. עבור כל איבר, נבדוק האם הוא בטבלה בזמן ממוצע $\Theta(n)$. סה"כ זמן ריצה $O(n^2)$ בממוצע.

7. **(8)** נתון מערך בגודל n. רוצים אלגוריתם המוצא שלשה של איברים במערך : x,y,z כך ש :

x + y = z. אם אין שלשה כזו, האלגוריתם מחזיר ׳לא׳.

הציעו את האלגוריתם המהיר ביותר (W.C) לפתרון הבעיה :

זמן האלגוריתם הוא :

א. $O(n\log n)$

VVב. $O(n^2)$

ג. $O(n^2\log n)$

ד. $O(n^3)$

ה. $O(n^3\log n)$

ו. $O(n^2\log^2 n)$

הסבר קצר : *נמיין את המערך, וניצור שני עותקים ממוינים שלו, שלב זה לוקח Θ(nlog n) . נבצע את השלב הבא עבור כל איבר X במערך:*
נוסיף X לכל איברי אחד המערכים (בזמן ליניארי) נמזג את שני המערכים בזמן ליניארי (לזכור כי הם ממוינים), ונבדוק אם יש איבר המופיע פעמיים. אם יש נחזיר TRUE.
כל השלב לוקח זמן ליניארי.

היות שאנו מבצעים את השלב n פעמים, וכל שלב לוקח זמן ליניארי, סה״כ הסיבוכיות היא $O(n^2)$.
ניתן לפתור את הנ״ל בעזרת עץ מאוזן בזמן w.c. O(n lg n).
תשובה כזו קיבלה ניקוד חלקי (4 נקודות).

8. **(4)** נתונה ערימת מקסימום בגודל n של מפתחות חיוביים. מפתח השורש הוא n. ידוע שהמפתח של כל צומת קטן פי 2 מהמפתח של אבא שלו. אזי בהכרח :

סכום כל המפתחות הוא $\Theta(n)$

א. נכון

VVב. לא נכון

הסבר קצר : *לא נכון. סכום המפתחות בכל שכבה בעץ הוא* $\Theta(n)$.
היות שלעץ Θ(log n) רמות, סכום המפתחות הוא Θ(nlog n) .

9. **(4)** נתונה ערימת מקסימום בגודל n של מפתחות חיוביים. מפתח השורש הוא n. ידוע שהמפתח של כל צומת קטן פי 2 מהמפתח של אבא שלו. אזי בהכרח:

סכום המפתחות של העלים הוא $\Theta(n)$

VVא. נכון

ב. לא נכון

הסבר קצר: *לערמה* $\Theta(n)$ *עלים. המפתח בכל עלה הוא* $\Theta(1)$*, לכל הסכום הוא* $\Theta(n)$*.*

10. (8) שינו את אלגוריתם MergeSort: בכל שלב, במקום לחלק את המערך לשני חלקים שווים, מגרילים מספר i בין 1 לגודל המערך פחות אחד (כאשר 1 מתקבל בהסתברות 2/n והשאר בהסתברות 1/n), ומחלקים את המערך לחלק אחד בגודל i וחלק שני - שאר האיברים. יתר האלגוריתם ללא שינוי. להלן Pseudo-code של האלגוריתם החדש:

```
MergeSort2(A,p,q)
    if p = q then return
    i = random number from (1…q – p )
    MergeSort2 (A, p, p + i - 1)
    MergeSort2 (A, p + i , q)
    Merge (A, p, p + i - 1, q)
```

זמן הריצה של האלגוריתם הנ"ל הוא:

אVV. ממוצע $\Theta(n\log n)$, גרוע ביותר $\Theta(n^2)$

ב. ממוצע $\Theta(n^2)$, גרוע ביותר $\Theta(n^2)$

ג. ממוצע $\Theta(n)$, גרוע ביותר $\Theta(n\log n)$

ד. ממוצע $\Theta(n)$, גרוע ביותר $\Theta(n^2)$

ה. ממוצע $\Theta(n\log n)$, גרוע ביותר $\Theta(n\log n)$

הסבר קצר: *הניתוח זהה לחלוטין לאלגוריתם QuickSort לכן תשובה א' נכונה.*

11. **(14)** נתונות n נקודות במישור. כל נקודה מיוצגת על ידי זוג סדור (x,y) (הנקודה (5,6) שונה מהנקודה (6,5)). הציעו מבנה נתונים התומך בפעולות :

Insert(x,y) – מוסיפה למבנה את הנקודה (x,y).

Delete(x,y) – מוחקת מהמבנה את הנקודה (x,y) אם היתה במבנה. אחרת לא עושה כלום.

Member(x,y) – מחזירה True אם הנקודה (x,y) במבנה. אחרת מחזירה False.

סיבוכיות ה-.W.C של כל הפעולות הנ״ל צריכה להיות O(log n)

א. **(7)**

תאור המבנה

Create a 2-3 tree where the key consists of the pair <x,y> in lexicographic order (x most significant).

Other solutions of same complexity are possible as well.

ניתוח זמן ריצה

Since this is a 2-3 tree, all operations will be conducted in O(logN).

מספר מחברת:

ב. (7) הציעו איך ניתן לממש את הנ"ל ובנוסף את הפעולה Larger(a,b) .
הפעולה הזו מדפיסה את כל הנקודות (x,y) שמקיימות $x>a$ **וגם** את כל הנקודות (x,y) שמקיימות $y>b$. זמן W.C. לפעולה הוא $O(k + \log n)$ כאשר k הוא מספר האיברים שיודפסו.
זמן הפעולות מהסעיף הקודם ללא שינוי.

תאור המבנה המשופר:

Construct the tree from the previous section. Also construct a similar tree where the key is <y,x> and the most significant key is y. For both trees construct a doubly linked list at the leaves.

ניתוח זמן ריצה:

All operations as before (for some of them need to conduct on both trees). For Large(a,b) go on the first tree to the largest element (O(logN)). Then go to the left on the linked list until finding x<a. Print all keys found on this leaves traversal (O(K)). Conduct a similar traversal on the second tree with the values b and y. The overall complexity is O(logN + K).

12. **(22)** נתונים שני עצי 2-3 T1 ו-T2 בגודל n1 ו-n2 בהתאמה.

א. **(7)** הציעו אלגוריתם שמאחד את שני העצים לעץ 2-3 חוקי בזמן O(n1+n2) W.C.

תאור מילולי של האלגוריתם (**לא** בפסאודו-קוד):

נסרוק כל עץ inorder לקבלת שני מערכים ממוינים.
נאחד את שני המערכים למערך ממוין (כמו בפעולת Merge של MergeSort).
נבנה עץ ריק המכיל n1+n2 עלים.
נסרוק את העץ Inorder ונעדכן את העלים ואת המפתחות.

ניתוח זמן הריצה:

כל אחת מהפעולות הנ"ל לוקחת זמן לינארי בגודל הקלט,
לכן סה"כ זמן ריצה O(n1+n2).

ב. **(7)** נניח שידוע שכל האיברים בעץ T1 קטנים מכל האיברים בעץ T2.

הציעו אלגוריתם שמאחד את שני העצים לעץ 2-3 חוקי בזמן W.C. O(log (Max(n1,n2))).

תאור מילולי של האלגוריתם (**לא** בפסאודו-קוד):

ניתלה את העץ הגבוה יותר על העץ הנמוך יותר:
נניח ש–T1 בגובה h1 ו–T2 בגובה h2 וכן h2>h1.
נרד ב–T2 שמאלה h2-h1 צעדים, ונתלה שם את T1.
אם לצומת שעליו תלינו היו מראש 3 בנים, אז נפצל אותו ונמשיך לתקן כלפי מעלה, כמו בפעולת insert רגילה.
אם h2<h1 הפתרון באופן דומה אבל יורדים ימינה על T1, h1-h2 צעדים. אם h1=h2 נגדיר להם שורש חדש משותף.

ניתוח זמן הריצה:

חיפוש המקום לתליה על העץ: O(|h1-h2|)
תליה: O(1)
תיקונים: O(|h1-h2|).
סה"כ: O(|h1-h2|) *ולכן גם* O(log(max(n1,n2)))
(לפי חסם על גובה עצי 2-3).

מספר מחברת:

ג. **(8)** הוכיחו שלא ניתן לפתור את סעיף א׳ בזמן $O(\sqrt{n1+n2})$ במודל ההשוואות.

הוכחה: נניח בשלילה כי קיים אלגוריתם כנ"ל. נגדיר את אלגוריתם המיון הבא: בהנתן n מספרים, חלק לשתי קבוצות, בנה מהן עץ 2-3 באופן רקורסיבי, מזג את שני העצים לעץ אחד. לפי הנחה, זמן הריצה מקיים $T(n)=2T(n/2)+O(n^{0.5})$. לפי שיטת האב $T(n)=\Theta(n)$. לכן לפי הנחה ניתן ליצור בזמן ליניארי עץ 2-3. בשלב זה ניתן לבצע סריקת INORDER ולקבל מיון של האיברים בזמן ליניארי, סתירה.

מספר תעודת זהות: 1

מספר מחברת:

מספר קורס: 0368.2158

סמסטר ב', מועד א', תשס"ב

21/7/2002

מבחן במבני נתונים

פרופ' יוסי מטיאס, מר עודד שוורץ

משך המבחן שלוש וחצי שעות (אין הארכה)

הבחינה עם חומר סגור, ללא מחשבים/מחשבונים.

<u>הוראות כלליות:</u>

1. יש לרשום מספר ת.ז. ומספר מחברת בראש כל דף בטופס זה.
2. יש לענות על כל השאלות.
3. **תשובות לשאלות הפתוחות יש לכתוב <u>על טופס המבחן בלבד</u>.** המחברות הן לטיוטה בלבד ולא תיבדקנה.
4. **תשובות לשאלות האמריקאיות יש לכתוב אך ורק <u>על הטופס המצורף</u>**. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
5. ניתן להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.

בהצלחה!!

<u>שיטת האב – Master Method</u>

מספר תעודת זהות: 3

מספר מחברת:

1. נציע אלגוריתם חדש למיון מערך :

אם גודל המערך לכל היותר 2 – מיין אותו בזמן קבוע.

אחרת :

מיין את שני השלישים הראשונים של המערך (ברקורסיה)

מיין את שני השלישים האחרונים של המערך (ברקורסיה)

מיין שוב את שני השלישים הראשונים של המערך

זמן הריצה של האלגוריתם הנ״ל הוא :

א. $\Theta(n\lg n)$

ב. $\Theta(n(\lg n)^{\frac{3}{2}})$

ג. $\Theta(n^{\frac{3}{2}}\lg n)$

ד. $\Theta(n^{\log_{\frac{3}{2}} 3})$

ה. $\Theta(n^{\log_{\frac{2}{3}} 3})$

ו. $\Theta(n)$

2.

$$f(n) = (\sqrt{\lg n})^n,\ g(n) = n^{\sqrt{\lg n}}$$

מה מהבאים נכון ?

א. f = Θ(g)

ב. f = o(g) (o קטן).

ג. g = o(f) (o קטן).

ד. אף אחד מהנ״ל

מספר תעודת זהות: ______

מספר מחברת: ______

3. נתונות שתי ערימות מקסימום במימוש מערך, המכילות n איברים כל אחת. רוצים לאחד אותן לערימת מקסימום יחידה, בזמן ריצה .W.C היעיל ביותר, במודל ההשוואות. הזמן הוא :

א. $\Theta(1)$

ב. $\Theta(\lg n)$

ג. $\Theta((\lg n)^2)$

ד. $\Theta(n)$

ה. $\Theta(n \lg n)$

4. נתון מבנה A שתומך בפעולות Insert, Delete, Find. אם נתון שזמן Amortized לפעולה הוא $\Theta(\lg n)$, אז יתכן שפעולת Insert לוקחת ב-.W.C :

א. $\Theta(\lg \lg n)$

ב. $\Theta(\lg n)$

ג. $\Theta(n)$

ד. א+ב

ה. ב+ג

ו. א+ב+ג

מספר תעודת זהות:

מספר מחברת:

5. נתון מערך בגודל n^3 שבו יש n^2 ערכים שונים. אלגוריתם למיון המערך, היעיל ביותר רץ בזמן W.C.:

א. $\Theta(n^2 \lg n)$

ב. $\Theta(n^3)$

ג. $\Theta(n^3 \lg n)$

ד. $\Theta(n^3 \lg n)$

ה. $\Theta(n^3 \lg^2 n)$

ו. $\Theta(n^3 \lg^3 n)$

6. נתון מספר k ומערך A בגודל n. המערך "כמעט ממוין" כלומר כל איבר נמצא במרחק לכל היותר $\lg n$ מקומות ממקומו במערך הממוין. רוצים אלגוריתם יעיל ביותר (ב-W.C.) לחיפוש הערך k במערך A. ניתן לעשות זאת בזמן:

א. $\Theta(n \lg n)$

ב. $\Theta(n)$

ג. $\Theta(\frac{n}{\lg n})$

ד. $\Theta((\lg n)^2)$

ה. $\Theta(\lg n)$

ו. $\Theta(\lg \lg n)$

7. רוצים אלגוריתם שמקבל קלט מערך בגודל n, ומדפיס את הסדרים הסטטיסטים $\frac{n}{2}, \frac{n}{2}+1, \frac{n}{2}+2, ..., \frac{n}{2}+\lg n,$ כלומר את החציון ואת $\lg n,$ האיברים שהיו באים אחריו לו המערך היה ממוין. אפשר לבצע זאת בזמן W.C. (היעיל ביותר):

א. $\Theta(n \lg n)$

ב. $\Theta(n)$

ג. $\Theta(\frac{n}{\lg n})$

ד. $\Theta((\lg n)^2)$

ה. $\Theta(\lg n)$

8. נתונים שני עצי חיפוש T1 ו-T2, שמכילים n מפתחות כל אחד. ידוע שכל המפתחות ב-T2 גדולים מכל המפתחות ב-T1. מחברים את העצים לעץ אחד בשם T באופן הבא:
יהי x צומת חדש, שיוגדר להיות השורש של העץ. T2 יחובר כבן ימני של x ו-T1 יחובר כבן שמאלי של x. ערך המפתח של x יוגדר להיות מספר כלשהו שגדול מכל המפתחות ב-T1 וקטן מכל המפתחות ב-T2.
באיזה מהמקרים הבאים מובטח שהעץ T הוא חוקי?

א. כשהעצים הם עצי חיפוש רגילים.

ב. כשהעצים הם עצי חיפוש AVL.

ג. כשהעצים הם עצי חיפוש 2-3.

ד. א + ב

ה. א + ג

ו. א + ב + ג

מספר תעודת זהות:

מספר מחברת:

9. נתונים n קטעים על הישר (על ידי נקודות התחלה וסיום). רוצים להכריע האם קיימים שני קטעים שנחתכים בנקודה אחת. (כלומר האם קיימת נקודה שמוכלת בלפחות שניים מ-n הקטעים הנ״ל).

טענה: ניתן לבצע זאת בזמן W.C. $O(n\lg\lg n)$ במודל ההשוואות.

מיחקו את המיותר:

הטענה נכונה / הטענה לא נכונה

הוכחה:

מספר תעודת זהות:

מספר מחברת:

10. רוצים מבנה נתונים שתומך בפעולות איתחול, הכנסה וחיפוש.

פעולת האיתחול מבוצעת על קבוצת מספרים בגודל n.

פעולת הכנסה של איבר תבוצע לכל היותר $\lg n$ פעמים.

דרישות מהמבנה :

זיכרון : $\Theta(n)$

זמן .W.C : פעולת הכנסה : $O(\lg \lg n)$

פעולת חיפוש : $O(\lg \lg n)$

פעולת איתחול : לא מוגבל

תאור המבנה :

הסבר פעולת איתחול :

הסבר פעולת חיפוש (כולל ניתוח זמן ריצה) :

גרסה א׳

מס סטודנט____________

מס מחברת ____________

מספר קורס: 0368.2158

סמסטר א׳, מועד א׳, תשס״ב **7.3.02**

מבחן במבני נתונים

פרופ׳ חנוך לוי, פרופ׳ עמוס פיאט, עדי אבידור, עודד שוורץ

הוראות כלליות:

1. יש לענות על כל השאלות.
2. משך המבחן **שלוש שעות וחצי**.
3. הבחינה עם **חומר סגור**, מותר להשתמש אך ורק בדף עזר אחד בגודל A4 הכתוב בכתב יד מצד אחד. אין שימוש בזכוכית מגדלת. אין שימוש בכל ציוד אלקטרוני.
4. **יש לכתוב תשובות סופיות לשאלות הפתוחות על טופס המבחן בלבד.**
5. **תשובות לשאלות האמריקאיות יש לכתוב אך ורק על הטופס המצורף.**
6. **במידה והתבקשתם להסביר / לנמק שאלה אמריקאית, יש לעשות זאת ברבוע ההערה השייך לשאלה בלבד.**
7. **במידה וברצונכם להוסיף הערה לבודק בנוגע לשאלה מסוימת, יש לעשות זאת ברבוע ההערה השייך לשאלה בלבד. הערות לבודק תיבדקנה רק במקרה של ערעור.**
8. המחברות הן לטיוטה בלבד. מחברות לא תיבדקנה.
9. ניתן להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
10. יש למצוא אלגוריתמים עם סיבוכיות טובה ככל האפשר. אלגוריתם יחשב יעיל יותר אם סיבוכיות ה- worst case שלו יעילה יותר מאלגוריתם אחר, או שהיא שווה לאלגוריתם האחר ויעילות ה - average שלו טובה יותר.
11. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
12. עליכם לצבור את **מרב הנקודות האפשרי**.

בהצלחה!!

גרסה א׳

מס סטודנט_____________

מס מחברת _____________

1. (6 נקודות) הינכם מקבלים בקלט N מפתחות, $x_1, x_2 \ldots, x_N$ שונים זה מזה ובסדר שרירותי (כלומר לא בהכרח בסדר $x_1, x_2 \ldots, x_N$). לאחר מכן הנכם מקבלים רצף של N בקשות למפתח x_1, N בקשות למפתח x_2,... ו-N בקשות למפתח x_N. הננו מעוניינים לבצע את העבודה של שליפת הנתונים במספר הקטן ביותר של פעולות ב- worst case (סה״כ עלות על כל השליפות!). הינכם יכולים לבחור את אחד ממבני הנתונים שנלמדו בהרצאה (לא בתרגול) ואסור לכם לשנותם בכלום. עליכם לבצע את כל השליפות מתוך מבנה הנתונים (לא ניתן לשמור תוצאות ביניים בצד). עלות ה-worst case הטובה ביותר שניתן להשיג היא:

א) $O(N^2 \log N)$ באמצעות עץ 2-3

ב) $O(N^2 \log^2 N)$ באמצעות עץ 2-3

ג) $O(N^2)$ ב-worst case, באמצעות טבלת hash עם open-addressing (אין הכוונה ל perfect hashing)

ד) $O(N^2)$ בממוצע, אבל $O(N^2 \log N)$ במקרה הגרוע באמצעות עץ חיפוש בינארי

ה) $O(N^2)$ באמצעות splay-tree

ו) $O(N^{5/2})$ באמצעות splay-tree

ז) $O(N^2 \log\log N)$ באמצעות עץ 2-3

ח) $O(N^2)$ בממוצע, אבל $O(N^3)$ ב-worst case, באמצעות טבלת hash עם closed-addressing (אין הכוונה ל perfect hashing)

הערות לבודק (רק אם יש!):

מס סטודנט______________

גרסה א'

מס מחברת ______________

2. התייחסו לעץ אדום שחור כפי שנלמד בכיתה ומתואר ב-Cormen (כל ה-Nils נחשבים כעלים וצבעם הוא שחור). יהי הגובה של צומת y מוגדר כאורך המסלול (נספר בצמתים) הארוך ביותר מעלה ל-y. הספירה לא כוללת את y ואת ה-nil הנידון. יהי x צומת <u>שמושמט</u> (**שימו לב: הכוונה לצומת המושמט ולא לערך המושמט**) בפעולת ה- delete (כמו שנלמד בכיתה בעץ חיפוש בינארי). הננו מעוניינים בגובהו של x לפני ההשמטה. אזי:

א) החסם העליון ההדוק ביותר על גובהו של x הינו 0 (כלומר גובה ≤ 0).
ב) החסם העליון ההדוק ביותר על גובהו של x הינו 1 (כלומר גובה ≤ 1).
ג) החסם העליון ההדוק ביותר על גובהו של x הינו 2 (כלומר גובה ≤ 2).
ד) החסם העליון ההדוק ביותר על גובהו של x הינו 3(כלומר גובה ≤ 3).
ה) אין חסם עליון על גובהו של x .

הערות לבודק (רק אם יש!):

3. (3 נקודות) משפחת פונקציות $H = \{h \mid h : M \to \{1\ldots,n\}\}$ נקראת אוניברסלית אם לכל x, y שונים ב-M, מספר הפונקציות h ב- H עבורם $h(x) = h(y)$ היא כגודלה של H חלקי n.

משפחת פונקציות $H = \{h \mid h : M \to \{1\ldots,n\}\}$ נקראת 2-אוניברסלית אם לכל x, y שונים ב-M, הזוג $\langle h(x), h(y)\rangle$ מתפלג אחיד מהקבוצה $\{\langle i, j\rangle \mid 1 \leq i, j \leq n\}$.

האם נכון שכל משפחה 2-אוניברסלית היא אוניברסלית?

א) כן
ב) לא

הערות לבודק (רק אם יש!):

מס סטודנט____________

גרסה א׳

מס מחברת ____________

4. (3 נקודות) משפחת פונקציות $H = \{h \mid h: M \to \{1\ldots,n\}\}$ נקראת אוניברסלית אם לכל x, y שונים ב-M, מספר הפונקציות h ב- H עבורם $h(x) = h(y)$ היא כגודלה של H חלקי n.

משפחת פונקציות $H = \{h \mid h: M \to \{1\ldots,n\}\}$ נקראת כמעט-אוניברסלית אם לכל x, y שונים ב-M, אם נבחר h באופן אחיד מתוך H אזי ההסתברות ש $h(x) = h(y)$ קטנה מ 2/n.

האם נכון שכל משפחה אוניברסלית היא כמעט אוניברסלית?

א) כן
ב) לא

הערות לבודק (רק אם יש!):

5. (6 נקודות) נתון מערך בגודל n שכל ערכיו מהקבוצה $\left\{1,2,\ldots,\left\lfloor n^{\log\log n}\right\rfloor\right\}$. אזי ניתן למיין את איברי המערך:

א) בזמן $\Theta(n)$ במקרה הממוצע
ב) בזמן $\Theta(n)$ במקרה הגרוע
ג) בזמן $\Theta(n\log n)$ במקרה הגרוע
ד) בזמן $\Theta(n\log\log n)$ במקרה הגרוע.
ה) בזמן $\Theta\left(n\cdot\frac{\log n}{\log\log n}\right)$ במקרה הגרוע
ו) בזמן $\Theta\left(n\cdot\frac{\log n}{\log\log n}\right)$ במקרה הממוצע

הערות לבודק (רק אם יש!):

גרסה א׳

מס סטודנט_____________

מס מחברת _____________

6. (6 נקודות) נתון מערך עם n איברים בו $n^{\frac{2001}{2002}}$ מהאיברים זהים. ניתן למיין את המערך בזמן worst case (יש לסמן את התשובה ההדוקה ביותר, בהנחת מודל RAM):

א) $\Theta(n)$

ב) $\Theta(n^{\frac{1}{2002}}\log n)$

ג) $\Theta(n\cdot(\log n)^{\frac{1}{2002}})$

ד) $\Theta(n^{\frac{2001}{2002}}\log n)$

ה) $\Theta(n\cdot(\log n)^{\frac{2001}{2002}})$

ו) $\Theta(n\log n)$

הערות לבודק (רק אם יש!):

7. (6 נקודות) נדפיס את איברי ערימת מקסימום עם n איברים בסדר **preorder** (שימו לב: **preorder=DLR**!). אזי בהנחה ש n גדול (לדוגמה n>2002), אילו מהבאים נכון:

א) איבר המינימום יכול להופיע במקום הראשון

ב) איבר המינימום תמיד יופיע באחד מ $\boldsymbol{\left\lfloor \frac{n}{2}\right\rfloor+1}$ המקומות האחרונים

ג) איבר המינימום יכול להופיע במקום ה $\lfloor \log n\rfloor+1$

ד) אם הערימה מלאה לגמרי גם בשכבה התחתונה אז איבר המינימום יופיע בדיוק במקום ה $\left\lfloor \frac{n}{2}\right\rfloor$

הערות לבודק (רק אם יש!):

מס סטודנט____________

מס מחברת ____________

8. (6 נקודות) בהינתן מבנה נתונים ריק וסדרה כלשהי של n פעולות עליו - זמן Amortized של פעולה במבנה הנתונים יוגדר כזמן הכולל של n הפעולות מחולק ב n.
נתונה מחסנית התומכת בפעולות:

- Pop() בזמן O(1)
- Push(x) בזמן O(1)
- PopLog() בזמן O(log n) - השולפת log n איברים מראש המחסנית (כאשר n הוא מספר האיברים הנוכחי במחסנית)

אזי בהינתן סדרה כלשהי של n פעולות המתחילה במחסנית ריקה, זמן Amortized של פעולה במחסנית הוא:

א) $\Theta(\log\log n)$
ב) $\Theta(n)$
ג) $\Theta\left(\log\frac{n}{\log n}\right)$
ד) $\Theta\left(\frac{n}{\log n}\right)$
ה) $\Theta(1)$
ו) $\Theta(\log n)$

הערות לבודק (רק אם יש!):

9. (6 נקודות) נתון מערך עם m איברים. n המקומות הראשונים מכילים את הערך 1 והשאר מכילים 0 (n לא ידוע). רוצים למצוא את n. ניתן לעשות זאת בזמן (worst case) היעיל ביותר:

א) $\Theta(1)$
ב) $\Theta(\log n)$
ג) $\Theta(\min\{\log m, n\})$
ד) $\Theta(m)$

הערות לבודק (רק אם יש!):

מס סטודנט_____________

גרסה א׳

מס מחברת _____________

10. (5 נקודות) נתון תור של מספרים טבעיים Q_1 עם n איברים. רוצים למיין את האיברים ב Q_1, כלומר בסיום ריצת אלגוריתם המיון Q_1 יכיל את אותם האיברים כמו בתחילת הריצה רק בסדר ממוין. מותר לאלגוריתם המיון להשתמש בתור אחד נוסף Q_2 עם $O(n)$ איברים ובעוד $O(1)$ זיכרון נוסף. אזי קיים אלגוריתם כנ״ל למיון

א) שזמן ריצתו $O(n^3)$ והמשתמש בעקרונות הדומים ל Bubble Sort

ב) שזמן ריצתו $O(n^2)$ והמשתמש בעקרונות הדומים ל Bubble Sort

ג) שזמן ריצתו $O(n^3 \log n)$ והמשתמש בעקרונות הדומים ל Quick Sort (הכוונה לגרסת Quick Sort בעלת זמן ריצה $O(n \log n)$ במקרה הגרוע)

ד) שזמן ריצתו $O(n^2 \log n)$ והמשתמש בעקרונות הדומים ל Quick Sort (הכוונה לגרסת Quick Sort בעלת זמן ריצה $O(n \log n)$ במקרה הגרוע)

ה) שזמן ריצתו $O(n^2 \log n)$ והמשתמש בעקרונות הדומים ל Merge Sort

ו) שזמן ריצתו $O(n \log n)$ והמשתמש בעקרונות הדומים ל Merge Sort

הערות לבודק (רק אם יש!):

מס סטודנט____________

גרסה א׳

מס מחברת ____________

11. (13 נקודות) רוצים לבנות מבנה נתונים לתחזוקת מספרים טבעיים התומך בפעולות insert(x), delete(x), find(x), עבור מספר טבעי x, ובפעולת multiple_of_5() המחזירה שני ערכים שונים במבנה x, y המקימים y=5x. אזי קיים מבנה כנ״ל בו (אנו מעוניינים במבנה התומך בפעולות עם סיבוכיות הטובה ביותר) (4 נקודות):

א) פעולות insert(x), delete(x) נעשות בזמן $O(\log^2 n)$ במקרה הגרוע, ופעולות find(x), multiple_of_5() נעשות בזמן $O(\log n)$ במקרה הגרוע.

ב) פעולות insert(x), delete(x) נעשות בזמן $O(\log^2 n)$ במקרה הגרוע, ופעולות find(x), multiple_of_5() נעשות בזמן $O(\log n)$ במקרה הגרוע.

ג) פעולות insert(x), find(x), delete(x) נעשות בזמן O(log n) במקרה הגרוע, ופעולת multiple_of_5() נעשית בזמן $O(n^2)$ במקרה הגרוע.

ד) פעולות insert(x), delete(x) נעשות בזמן $O(\log^2 n)$ במקרה הגרוע, פעולת find(x) נעשית בזמן $O(\log n)$ ופעולת multiple_of_5() נעשית בזמן $O(n)$ במקרה הגרוע.

ה) פעולות insert(x), find(x), delete(x) נעשות בזמן O(log n) במקרה הגרוע, ופעולת multiple_of_5() נעשית בזמן O(1) במקרה הגרוע.

ו) פעולות insert(x), find(x), delete(x) נעשות בזמן O(log n) במקרה הגרוע, ופעולת multiple_of_5() נעשית בזמן O(n) במקרה הגרוע.

ז) כל הפעולות נעשות בזמן O(log n) במקרה הגרוע.

תארו בקצרה את מבנה הנתונים ואת הפעולות (9 נקודות):

התשובה הנכונה היא ה׳.
מבנה הנתונים יהיה עץ 2-3 (או עץ חיפוש מאוזן אחר) שמכיל את כל האיברים. רשימה דו כיוונית שמכילה את הזוגות שהמנה ביניהם היא 5. כמו כן יש הצבעות הדדיות בין האיברים בשני המבנים.
פעולות הוספה, מחיקה וחיפוש מבוצעות על העץ. במקרה של הוספה ומחיקה יש לעדכן גם את הרשימה (כלומר להוסיף או להוריד זוג מהרשימה, לפי הצורך).
פעולת Multiple_of_5() תבוצע על ידי החזרת ראש הרשימה המקושרת.

מבני נתונים 2 מתוך5

מס סטודנט____________

גרסה א׳

מס מחברת ____________

2. (10 נקודות) זוג ערכים במערך נקראים עוקבים אם הם שונים ולאחר **מיון** הם נמצאים במקומות סמוכים במערך. בהינתן מערך של מספרים טבעיים בגודל n, הציעו אלגוריתם (בעל סיבוכיות worst case הטובה ביותר) הבודק האם יש זוג ערכים עוקבים שסכום **המופעים** של שניהם ביחד הוא לפחות $\left\lceil \frac{n}{2} \right\rceil$ (כלומר מספר הפעמים שמופיע ערך מסוים ועוד מספר הפעמים שמופיע הערך העוקב לו יהיה לפחות $\left\lceil \frac{n}{2} \right\rceil$).

תיאור האלגוריתם (8 נקודות):

ניתן לפתור ב- O(n) Worst case **בכמה דרכים** בהתבססות על order statistics.

למשל, פתרון שמתבסס על **<u>העקרונות</u>** הבאים (בלא פרוט פרטים טכניים):

1. מציאת חציון – O(n) (סמנו ב-M)
2. מציאת האיבר הקטן ביותר הגדול מהחציון (H) (אם יש) –O(n)
3. מציאת האיבר הגדול ביותר הקטן מהחציון (L) (אם יש)–O(n)
4. ספירת הופעות של M, L, H – O(n)
5. אם כמות ההופעות הכוללת של M ו-H או של M ו-L מקיימת את התנאי התשובה חיובית. אחרת שלילית.

מה סיבוכיות האלגוריתם (1 נקודה)?

נמקו בקצרה (1 נקודות):

מבני נתונים 3 מתוך5
מס סטודנט_____________
גרסה א'
מס מחברת _____________

3. (25 נקודות) נתונים מבני הנתונים הבאים:

a. רשימה מקושרת חד כיוונית לא ממוינת
b. רשימה מקושרת דו כיוונית ממוינת
c. Heap כשראש ה-Heap הוא האיבר המינימלי
d. Heap כשראש ה-Heap הוא האיבר המקסימלי
e. טבלת Hash הממומשת בעזרת רשימות מקושרות, כשנתון שפונקצית ה - Hash היא פונקציה רנדומלית מהתחום לטווח, גודל הטבלה n (כמספר האיברים המקסימלי בכל רגע נתון), כל כניסה בטבלה היא או Nil או מצביע לרשימה מקושרת דו כיוונית **ממוינת** של המפתחות שמופו לאותה כתובת. הניחו שפונקצית ה- Hash נבחרה רנדומלית ללא תלות במפתחות הקלט.

סדרת הפעולות שנבצע הם n פעולות Insert, אח"כ n פעולות Search לאיברים שהכנסנו, פעולת Minimum, פעולת Maximum, ואח"כ n פעולות Delete איברים שהכנסנו.

א. (16 נקודות) מלאו את הטבלה הבאה בזמני ריצה. השתמשו ב- O גדול - O(f(n)) בכל כניסה בטבלה. חלק מהכניסות כבר מולאו (כל כניסה תזכה בנקודה):

<u>**worst case** לפעולה יחידה</u>

	מבנה **a**	מבנה **b**	מבנה **c**	מבנה **d**
Search(key)	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Insert(key)	$O(1)$	$O(n)$	$O(\log n)$	$O(\log n)$
Delete(key)	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Minimum	$O(n)$	$O(1)$	$O(1)$	$O(n)$
Maximum	$O(n)$	$O(1)$	$O(n)$	$O(1)$

ב. (9 נקודות) סעיף זה מתייחס לטבלת ה Hash.

ה- expected average case וה- expected worst case מוגדרים מעל מרחב המדגם של בחירה אקראית יוניפורמית של פונקציה מתוך כל הפונקציות האפשריות ממרחב המפתחות לטווח הכתובות (ב expected worst case מסתכלים על הממוצע על פני כל הפונקציות של אורך הרשימה הארוכה ביותר בטבלה, וב expected average case מסתכלים על הממוצע על פני כל הפונקציות של האורך הממוצע של רשימה בטבלה).

כשמדובר ב- Minimum ו-Maximum אין משמעות לממוצע כי יש רק פעולה אחת כזו בסדרת הפעולות שלנו.

מלאו את הטבלה הבאה בזמני ריצה. השתמשו ב- O גדול - O(f(n)) בכל כניסה בטבלה. חלק מהכניסות כבר מולאו (כל כניסה תזכה בנקודה):

	e. hash expected average case	**e. hash expected worst case**
Search(key)	$O(1)$	$O(\log n)$
Insert(key)	$O(1)$	$O(\log n)$
Delete(key)	$O(1)$	$O(\log n)$
Minimum	$O(n)$	$O(n)$
Maximum	$O(n)$	$O(n)$

הערות לבודק (רק אם יש!):

מבני נתונים 5 מתוך 5

מס סטודנט______________

גרסה א'

מס מחברת ______________

תשובות לשאלות 1 עד 10 :

שאלה	גירסה א'	גירסה ב'	גירסה ג'	גירסה ד'
1	ה	ב	ד	ה
2	ב	ד	א	ה
3	א	א	א	ד
4	א	א	ב	א
5	ד	ו	ג	א
6	ו	ג	ו	ג
7	ג	ה	ה	ב
8	ה	ה	ב	ו
9	ב	ב	ו	ו
10	ו	ו	ה	ב

מס סטודנט____________
גרסה א׳
מס מחברת ____________

מספר קורס: 0368.2158

מועד ב', תש"סא

מבחן במבני נתונים

מר ערן הלפרין, מר עודד שוורץ

הורואת כלליות:

1. יש לענות על כל השאלות.
2. אורך המבחן שלוש שעות.
3. הבחינה עם חומר סגור, ללא מחשבים/מחשבונים.
4. **יש לכתוב תשובות סופיות לשאלות הפתוחות על טופס המבחן בלבד.**
5. **תשובות לשאלות האמריקאיות יש לכתוב אך ורק על הטופס המצורף**.
6. **במידה וברצונכם להוסיף הערה לבודק בנוגע לשאלה מסוימת, יש לעשות זאת ברבוע ההערה השייך לשאלה**. הערות יבדקו רק במקרה של עירעור.
7. המחברות הן לטיוטה בלבד. מחברות לא תיבדקנה.
8. ניתן להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
9. יש למצוא אלגוריתמים עם סיבוכיות טובה ככל האפשר. אלגוריתם יחשב יעיל יותר אם סיבוכיות ה- Worst case שלו יעילה יותר מאלגוריתם אחר,

מס סטודנט____________
גרסה א׳
מס מחברת ____________

1. בסדרה של N פעולות Insert לעץ 2-3 (ריק מלכתחילה):
יהא F(N) מספר פעולות הוספת צומת חדש (new) הכולל במקרה הגרוע ביותר.
יהא G(N) מספר פעולות המחשב הכולל במקרה הגרוע ביותר.

מה החסם העליון ההדוק ביותר ליחס $\frac{G(N)}{F(N)}$?

א. θ(1)

ב. θ (lg(N))

ג. θ (N)

ד. θ (Nlg(N))

ה. θ (N/lg(N))

הערה לבודק (במידה ויש!) עבור שאלה 1:

2. נתונה ערמת מינימום (heap min) בגודל N. רוצים להופכה לעץ אדום שחור. ניתן לבצע זאת בזמן worst case של:

א. θ (lg(N))

ב. θ (N)

ג. θ (Nlg(N))

ד. θ (Nlg(N)lg(N))

ה. אף תשובה אינה נכונה (למעט ה').

הערה לבודק (במידה ויש!) עבור שאלה 2:

3. נתון מבנה Heap min **שרירותי** בגודל N. עבור K שרירותי רוצים למצוא את האיבר ה-K הקטן ביותר **ו**את האיבר ה-K הגדול ביותר . זמן הריצה ב-worst case, של האלגוריתם הטוב ביותר למציאת **שני האיברים האלו** בתלות ב-K **ו**-N, הינו:

א. θ (lg(N))

ב. θ (N)

ג. θ (Nlg(N))

ד. θ (Klg(K))

ה. θ (MIN{N, Klg(N)})

ו. θ ((N-K)lg(N-K))

הערה לבודק (במידה ויש!) עבור שאלה 3:

מס סטודנט____________
גרסה א'
מס מחברת ____________

4. נתון מערך בגודל n^3 שבו כל ערך מופיע בדיוק פעמיים. האלגורית היעיל היותר למיון המערך, במודל ההשוואות, פועל במקרה הגרוע בזמן:

א. $\Omega(n \lg(n))$

ב. $\Omega(n^3 \lg(n))$

ג. $\Omega(n^3 \lg^3(n))$

ד. $\Omega((n \lg(n))^3)$

ה. אף אחד מהנ"ל.

הערה לבודק (במידה ויש!) עבור שאלה :

5. ממשים תור בעזרת שתי מחסניות, כאשר כל מחסנית ממומשת בעזרת תור קדימויות (ערימה). מבצעים סידרה של n פעולות הכנסה או הוצאה מהתור (בסדר כלשהו) כשבהתחלה מבנה הנתונים ריק.

הזמן הממוצע לפעולה בודדת (במימוש יעיל ככל האפשר) הוא:

א. O(1)

ב. O(lg(n))

ג. O(n)

ד. O(n lg(n))

ה. אף אחד מהנ"ל.

הערה לבודק (במידה ויש!) עבור שאלה :

6. נתונים n קטעים [a_i, b_i]. (לכל i מתקיים $a_i < b_i$).
ידוע שלכל קטע b_i הוא מספר שלם בתחום $1...n^3$ (אבל זה לא בהכרח נכון עבור המספרים a_i).
רוצים להכריע האם כל הקטעים זרים. ניתן לבצע זאת בזמן

א. $O(n)$
ב. $O(n \lg(n))$
ג. $O(n \lg^3(n))$
ד. $O(n^3 \lg^3(n))$
ה. אף אחד מהנ"ל.

הערה לבודק (במידה ויש!) עבור שאלה :

7. "מודל ההשוואות +" הוא כמו מודל ההשואות, אלא שבהשוואת זוג איברים a ו-b אפשר לקבל אחת מחמש מהתשובות הבאות:

a = b

a גדול מעשרה איברים (או יותר) שגדולים מ-b.

b גדול מעשרה איברים (או יותר) שגדולים מ-a.

a גדול מ-b אבל יש פחות מעשרה איברים שערכם בין a ל-b.

b גדול מ-a אבל יש פחות מעשרה איברים שערכם בין a ל-b.

חסם תחתון ל-.W.C של אלגוריתם מיון במודל ההשוואות + הוא:

א. $\Omega(1)$

ב. $\Omega(\lg(n))$

ג. $\Omega(n)$

ד. $\Omega(n \lg(n))$

ה. אף אחד מהנ"ל.

הערה לבודק (במידה ויש!) עבור שאלה :

8. מכניסים 2n איברים לטבלת hash עם 3n תאים. בכל תא מקום לאיבר בודד. במקרה של התנגשות (מיפוי איבר לתא תפוס) מכניסים את האיבר החדש לרשימה מקושרת משותפת לכל האיברים.

תוחלת זמן חיפוש כושל במבנה (כלומר חיפוש ערך x שלא קיים במבנה) היא:

א. O(1)

ב. O(n)

ג. O(lg(n))

ד. אף אחד מהנ"ל.

הערה לבודק (במידה ויש!) עבור שאלה :

מס סטודנט____________

גרסה א׳

מס מחברת _____________

מספר קורס: 0368.2158

סמסטר ב׳, מועד א׳, תש״סא

מבחן במבני נתונים
מר ערן הלפרין, מר עודד שוורץ

הוראות כלליות:

1. יש לענות על כל השאלות.
2. אורך המבחן שלוש שעות.
3. הבחינה עם חומר סגור, ללא מחשבים/מחשבונים.
4. **יש לכתוב תשובות סופיות לשאלות הפתוחות על טופס המבחן בלבד.**
5. **תשובות לשאלות האמריקאיות יש לכתוב אך ורק על הטופס המצורף**.
6. המחברות הן לטיוטה בלבד. מחברות לא תיבדקנה.
7. ניתן להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
8. יש למצוא אלגוריתמים עם סיבוכיות טובה ככל האפשר. אלגוריתם יחשב יעיל יותר אם סיבוכיות ה- Worst case שלו יעילה יותר מאלגוריתם אחר, או שהיא שווה לאלגוריתם האחר ויעילות ה - average שלו טובה יותר.
9. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.

בהצלחה!!

מס סטודנט____________
גרסה א׳
מס מחברת ____________

1. בסדרה של N פעולות Insert לעץ אדום שחור (ריק מלכתחילה):
יהא F(N) מספר פעולות הרוטציה הכולל במקרה הגרוע ביותר.
יהא G(N) מספר פעולות המחשב הכולל במקרה הגרוע ביותר.

מה החסם העליון ההדוק ביותר ליחס $\frac{G(N)}{F(N)}$?

א. θ(1)

ב. θ (lg(N))

ג. θ (N)

ד. θ (Nlg(N))

2. נתון עץ בינארי בגודל N. רוצים לוודא שהוא עץ חיפוש. סיבוכיות ה-WC של האלגוריתם היעיל ביותר לביצוע הבדיקה הוא

א. θ (lg(N))

ב. θ (N)

ג. θ (Nlg(N))

ד. תלוי בגובה העץ.

3. מערך בעל n איברים יקרא "מוגזם" אם יש בו שלושה ערכים שכל אחד מהם מופיע לפחות $\frac{n}{3}(1-\frac{1}{\lg(n)})$ פעמים, ושאר הערכים מופיעים לכל היותר פעמיים כל אחד. מהו החסם התחתון של ה-Worst Case של אלגוריתם מיון של מערך "מוגזם"?

א. θ (n)

ב. θ (n lg lg (n))

ג. θ (n lg (n))

ד. $\Theta(\frac{n \cdot \lg\lg(n)}{\lg(n)})$

4. נתונות שתי ערמות מקסימום (heap max) בגודל N. רוצים לאחד אותן לערימה חוקית אחת. ניתן לבצע זאת בזמן:

א. θ (lg(N))

ב. θ (N)

ג. θ (Nlg(N))

ד. אף תשובה אינה נכונה (למעט ד').

5. נתונה טבלת Hash (עם פונקצית Hash שנבחרת באופן אקראי מקבוצה אוניברסלית). בטבלה זו התנגשויות נפתרות באופן הבא:
כל איבר שממופה לתא שאינו ריק, נשמר בעץ אדום שחור **משותף לכל הטבלה**. חיפוש מתבצע על ידי בדיקת התא המתאים (באמצעות פונקצית ה-Hash), ואם התא **אינו** ריק, והאיבר הרצוי **אינו** נמצא שם – מחפשים בעץ.

יהא B גודל הטבלה, N – מספר האיברים ו- $\alpha=N/B$.
זמן חיפוש כושל הוא:

א. $\Theta(1+\alpha)$ בתוחלת אם $B=\Theta(N)$.

ב. $\Theta(1)$ בתוחלת אם $N^2=B$; $\Theta(\log N)$ בתוחלת אם $10\cdot N=B$.

ג. $\Theta(1)$ ב-worst case אם $N^2=B$ (ה- worst case הוא על כל הסדרות של N איברים ובחירות הפונקציות); $\Theta(\log N)$ ב-worst case אם $10\cdot N=B$.

ד. $\Theta(\frac{\log(N-B)}{B})$ בתוחלת.

6. נתון מערך עם N איברים. ידוע שערך אחד מופיע בו בדיוק N/2 פעמים, ערך נוסף בדיוק N/4 פעמים, ערך נוסף N/8 פעמים וכו'. מה זמן ריצת אלגוריתם מיון (במודל ההשוואות) היעיל ביותר למערך זה (WC).

א. $\theta(\lg(N))$

ב. $\theta(N)$.

ג. $\theta(N\lg\lg(N))$

ד. $\theta(N\lg(N))$

7. הפתרון של נוסחת הנסיגה הבאה:

$$T(n) = \sqrt{n}T(\sqrt{n}) + 10$$

הוא

א. $\theta(n)$

ב. $(n^{1/2} \lg(n))$

ג. $(n \lg(n))$

ד. $\theta(n^{\lg(n)})$

8. מי יותר גדול?

$$f(n) = n^{10\lg(n)}, g(n) = (\lg(n))^n$$

א. $f(n)=O(g(n))$

ב. $f(n)=\Omega(g(n))$

ג. $f(n)=\Theta(g(n))$

ד. $f(n)=o(g(n))$

ה. $f(n) = \omega(g(n))$

9. נתונה ערימה (heap-max) בגודל N שבה כל הערכים שונים זה מזה.
הפעולה DECREASE_KEY(P, VAL) מקבלת כקלט מצביע P לאחד מאברי הערימה וערך VAL. התוצאה המתקבלת הינה ערימה אשר בה, יורד ערכו של האיבר אליו הצביע P.
לדוגמא, אם P הצביע לאיבר עם ערך 12 ערכו החדש יהיה 12-VAL.
בהנתן כי P מצביע לאיבר ה-K-י בגודלו בערימה (יש בדיוק K-1 איברים קטנים ממנו), באיזו יעילות ניתן לבצע את DECREASE_KEY(P, VAL) ?

א. θ (lg(k))

ב. θ (lg (n))

ג. θ (lg(n/k))

ד. θ (min(k,log n))

10. נגדיר מרחק בין שני צמתים בעץ כאורך המסלול הקצר ביותר ביניהם.
מהו המרחק המקסימלי האפשרי בין זוג צמתים בעץ אדום שחור בעל N צמתים?

א. $LOG_2(N) \pm \Theta(1)$

ב. $2LOG_2(N) \pm \Theta(1)$

ג. $3LOG_2(N) \pm \Theta(1)$

ד. $4LOG_2(N) \pm \Theta(1)$

11. נתון מערך שבו ה-OS(i) (האיבר ה-i בגודלו) נמצא באחד המקומות
i-k,i-k+1,….,i+k-1,i+k.

א. מצאו אלגוריתם יעיל ככל הניתן למיון מערך זה.

סיבוכיות האלגוריתם:

תיאור האלגוריתם:

ב. מצאו חסם תחתון במודל ההשוואות לאלגוריתמי מיון שהקלט שלהם הוא כמו בסעיף א'. הוכיחו את נכונות החסם התחתון.

גודל החסם:

הוכחה:

מס סטודנט___________

גרסה א׳

מס מחברת ___________

מספר קורס: 0368.2158

מועד ב', תש"סא

<u>מבחן במבני נתונים</u>

<u>פרופ. חנוך לוי, ד"ר טובה מילוא, מר ערן הלפרין, מר עודד שוורץ</u>

<u>הוראות כלליות:</u>

א. יש לענות על כל השאלות.
ב. אורך המבחן שלוש שעות.
ג. הבחינה עם חומר סגור, ללא מחשבים/מחשבונים.
ד. יש לכתוב תשובות סופיות לשאלות הפתוחות על טופס המבחן בלבד. המחברות הן לטיוטה בלבד. מחברות לא תיבדקנה.
ה. תשובות לשאלות האמריקאיות יש לכתוב אך ורק על הטופס המצורף.
ו. ניתן להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
ז. יש למצוא אלגוריתמים עם סיבוכיות טובה ככל האפשר. אלגוריתם יחשב יעיל יותר אם סיבוכיות ה- Worst case שלו יעילה יותר מאלגוריתם אחר, או שהיא שווה לאלגוריתם האחר ויעילות ה - average שלו טובה יותר.
ח. בכל שאלה אמריקאית יש תשובה אחת נכונה ביותר. רק תשובה זו תחשב לנכונה.
ט. בעמוד האחרון יש מקום להערות שלכם (אם יש) שיבדקו רק במקרה של ערעור.

בהצלחה!!

1. בסדרה של N פעולות Insert לעץ 2-3 (ריק מלכתחילה)
יהא F(N) מספר הפעמים שבוצע פיצול של צומת בעץ בהכנסת N איברים, במקרה הגרוע ביותר.
יהא G(N) מספר פעולות המחשב הכולל, בהכנסת N איברים, במקרה הגרוע ביותר.

מה החסם העליון ההדוק ביותר ליחס $\frac{G(N)}{F(N)}$?

א. $\theta(1)$
ב. $\theta(Log(N))$
ג. $\theta(N)$
ד. $\theta(NLog(N))$
ה. $\theta(N/Log(N))$

2. נתונה ערמת מינימום (heap min) בגודל N. רוצים להופכה לעץ אדום שחור. ניתן לבצע זאת בזמן worst case של:
א. $\theta(Log(N))$
ב. $\theta(N)$
ג. $\theta(NLog(N))$
ד. $\theta(NLog(N)Log(N))$
ה. אף תשובה אינה נכונה (למעט ה').

3. נתון מבנה Heap min **שרירותי** בגודל N. עבור K שרירותי רוצים למצוא את האיבר ה-K הקטן ביותר ואת האיבר ה-K הגדול ביותר . זמן הריצה ב-worst case, של האלגוריתם הטוב ביותר למציאת **שני האיברים האלו** בתלות ב-K ו-N, הינו:

א. θ (Log(N))

ב. θ (N)

ג. θ (NLog(N))

ד. θ (KLog(K))

ה. θ (MIN{N, KLog(N)})

ו. θ ((N-K)Log(N-K))

4. נתון מערך בגודל n^3 שבו כל ערך מופיע בדיוק פעמיים. האלגוריתם היעיל היותר למיון המערך, במודל ההשוואות, פועל במקרה הגרוע בזמן:

א. $\Omega(n \, Log(n))$

ב. $\Omega(n^3 \, Log(n))$

ג. $\Omega(n^3 \, Log^3(n))$

ד. $\Omega((n \, Log(n))^3)$

ה. אף אחד מהנ"ל.

5. "מודל ההשוואות +" דומה למודל ההשוואות, אלא שבהשוואת זוג איברים a ו-b אפשר לקבל אחת מחמש מהתשובות הבאות:

a = b

a גדול מעשרה איברים (או יותר) שגדולים מ-b.

b גדול מעשרה איברים (או יותר) שגדולים מ-a.

a גדול מ-b אבל יש פחות מעשרה איברים שערכם בין a ל-b.

b גדול מ-a אבל יש פחות מעשרה איברים שערכם בין a ל-b.

חסם תחתון ל-.W.C של אלגוריתם מיון במודל ההשוואות + הוא:

א. $\Omega(1)$

ב. $\Omega(Log(n))$

ג. $\Omega(n)$

ד. $\Omega(n\ Log(n))$

ה. אף אחד מהנ"ל.

6. מכניסים 2n איברים לטבלת hash עם 3n תאים. בכל תא מקום לאיבר בודד. במקרה של התנגשות (מיפוי איבר לתא תפוס) מכניסים את האיבר החדש לרשימה מקושרת משותפת לכל האיברים.

תוחלת זמן חיפוש כושל במבנה (כלומר חיפוש ערך x שלא קיים במבנה) היא:

א. $\Theta(1)$

ב. $\Theta(n)$

ג. $\Theta(log(n))$

ד. אף אחד מהנ"ל.

7. נתון מערך ממוין באורך n. בכמה זמן במקרה הגרוע ניתן להפוך את המערך לערימה במימוש מערך?

א. Θ (1)

ב. Θ (n)

ג. Θ (Log(n))

ד. Θ (nLog(n)).

8. נתון:

$$f\,(n) = 2^{Log_3 n}, g(n) = n^2$$

מהו היחס ההדוק ביותר?

א. g(n) = θ(f(n)).

ב. g(n) = o(f(n)).

ג. g(n) = ω(f(n)).

ד. g(n) = O(f(n)).

ה. g(n) = Ω(f(n)).

9. פתרו את נוסחת הנסיגה הבאה:

$$T(n) = 2T(\frac{n}{2}) + nLog^2(n) \quad , \quad T(n) = 10 \;\; for \;\; n \le 100$$

א. $T(n) = \theta(n)$
ב. $T(n) = \theta(nLog^2n)$
ג. $T(n) = \theta(nLog^3n)$
ד. $T(n) = \theta(n^2)$
ה. אף אחד מהנ"ל.

10. נתונה המחרוזת הבאה : "BAOBAB".
רוצים לרשום אותה בעזרת קוד הופמן, אשר מוגדר על האותיות המופיעות במחרוזת הנ"ל, לפי תדירות הופעתן במחרוזת.
אורך הקידוד (בביטים) יהיה:
א. 6
ב. 9
ג. 10
ד. 12
ה. אף אחד מהנ"ל.

11. לקחו עץ בינארי מסודר (יש חשיבות לסדר הבנים משמאל לימין, ואם יש בן יחיד, יש חשיבות אם הוא בן שמאלי או ימני). העץ מכיל n צמתים שממוספרים במספרים 1 עד n בסדר כלשהו. שימו לב שהעץ אינו בהכרח עץ חיפוש. ייצרו מעץ זה שני מערכים כמתואר להלן:

מערך באורך n המתאר סדר DLR של העץ.

מערך בגודל n, אשר מכיל במקום H(k) את עומק הצומת המסומן ב-k (למשל, אם k הוא השורש, H(k) = 0).

השאלה: האם בהינתן שני המערכים הנ"ל בלבד, ניתן לשחזר ביחידות את העץ?

מחקו את המיותר: לא ניתן לביצוע / ניתן לביצוע בזמן ______ (השלימו את החסר)

הסבר:

12. נתונים n קטעים [a_i, b_i]. (לכל i מתקיים $a_i < b_i$).
ידוע שלכל קטע b_i הוא מספר שלם בתחום $1...n^2$ (אבל זה לא בהכרח נכון עבור המספרים a_i).
רוצים להכריע האם כל הקטעים זרים.
מצאו אלגוריתם יעיל ככל האפשר לפתרון הבעיה הנ"ל.

סיבוכיות האלגוריתם:

תאור האלגוריתם:

מס מחברת _____________

מספר קורס: 0368.2158

סמסטר א׳, מועד א׳, תש״סא
תאריך הבחינה: 1.3.01

מבחן במבני נתונים
פרופ׳ חנוך לוי, דר׳ טובה מילוא, מר עודד שוורץ, מר ערן הלפרין

הוראות כלליות:

1. במבחן 5 שאלות. יש לענות על כולן.

2. יש למלא בכל דף את מספר הסטודנט ומספר המחברת. מלא/י עכשיו...

3. אורך המבחן שלוש שעות.

4. הבחינה עם חומר סגור. מותר להביא רק דף פוליו אחד. ללא מחשבים/מחשבונים.

5. **יש לכתוב התשובות רק על טופס המבחן**. מחברות לטיוטה בלבד. מחברות לא תיבדקנה.

6. ניתן להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.

7. יש למצוא אלגוריתמים עם סיבוכיות טובה ככל האפשר. אלגוריתם יחשב יעיל יותר אם סיבוכיות ה- Worst case שלו יעילה יותר מאלגוריתם אחר, או שהיא שווה לאלגוריתם האחר ויעילות ה - average שלו טובה יותר.

8. עליכם להשיג את מירב הנקודות בבחינה. ציון הבחינה יחושב על פי הנקודות שתצברו.

בהצלחה!!

שאלה 1 (18 נק׳)

פתרו את נוסחת הנסיגה הבאה:

$T(r,s) = r$ if $s < 10$ and $r \geq 10$

$T(r,s) = s$ if $r < 10$ and $s \geq 10$

$T(r,s) = 5$ if $r < 10$ and $s < 10$

$T(r,s) = r \cdot s + 4 \cdot T(½r,½s)$ Otherwise

פתרון הנוסחה:

$T(r,s) = \Theta(\qquad)$

נמוק:

שאלה 2 (18 נק׳)

נתונה פרוצדורה MEDIAN(A) המקבלת מערך A באורך n ומוצאת את החציון בזמן O(n).
השתמשו בפרוצדורה זו כקופסה שחורה על מנת ליישם פרוצדורה ORDER_STAT(A,k),
המקבלת מערך A באורך n ומספר שלם k, ומחזירה את ה-order statistics ה-k-י (האיבר ה-k-י בגודלו). **כמובן, אין להשתמש בפרוצדורה select (מציאת order statistics) שנלמדה בכיתה.**

סבוכיות worst case:

תיאור האלגוריתם:

הוכחת הנ

שאלה 3 (28 נק')

נתונים N זוגות של מספרים ממשיים, כאשר כל זוג ניכתב כ- (a,b) ומייצג קטע על הישר (ניתן להניח a<b).

נתון מספר ממשי x . ניתן להניח ש-x שונה מכל קצוות הקטעים (לכל (a,b), מתקיים a≠x , b≠x).
יהא La(x) מספר הזוגות המקיימים a<x ויהא Lb(x) מספר הזוגות המקיימים b<x.
יהא Ra(x) מספר הזוגות המקיימים x<a ויהא Rb(x) מספר הזוגות המקיימים x<b.

יהא CUT(x) מספר הקטעים (a,b) כך ש- x מוכל בהם (כלומר a<x<b).

דוגמה: נניח שהזוגות הם (1.2, 6.8), (2, 9), (10, 11.3) ן x=7 . אזי La(x)=2, Ra(x)=1 וכן הלאה.

א. (6 **נקודות**) בהינתן המשתנים La(x), Lb(x), Ra(x), Rb(x), הננו מעוניינים לחשב את CUT(x). ספקו נוסחה לחישוב CUT(x) כביטוי של המשתנים הנ"ל.

הנוסחה:

הסבירו בקצרה:

ב. (9 נק') תארו מבנה נתונים המתחזק קבוצת זוגות של מספרים ממשיים ומסוגל לבצע ביעילות את הפעולות הבאות (**הניחו שכל הזוגות שונים זה מזה**, כלומר שאין שני זוגות שבעבורם ערכי ה-a שווים ואין שני זוגות שעבורם ערכי ה-b שווים):

1. INSERT(a,b) - מכניס את הזוג (a,b) לתוך המבנה. ניתן להניח כי a < b.
2. DELETE(a,b) - מוציא את הזוג (a,b) מהמבנה (אם הוא שם מלכתחילה).
3. La(x) - מחזיר את מספר הזוגות (a,b) במבנה, כך ש-a < x. (גם כאן ניתן להניח כי x שונה מכל קצוות הקטעים (לכל a,b , מתקיים a≠x , b≠x).

סבוכיות INSERT	
סבוכיות DELETE	
סבוכיות La(x)	

תאור מבנה הנתונים:

ג. (13 נק') תארו מבנה נתונים המתחזק זוגות של מספרים ממשיים ומסוגל לבצע ביעילות את הפעולות הבאות (**הניחו שכל הזוגות שונים זה מזה**, כלומר שאין שני זוגות שבעבורם ערכי ה-a שווים ואין שני זוגות שעבורם ערכי ה-b שווים) :

1. INSERT(a,b) - מכניס את הזוג (a,b) לתוך המבנה. ניתן להניח כי $a < b$.
2. DELETE(a,b) - מוציא את הזוג (a,b) מהמבנה (אם הוא שם מלכתחילה).
3. CUT(x)- מחזיר את מספר הזוגות (a,b) במבנה, כך ש $a < x < b$. (גם כאן ניתן להניח כי x שונה מכל קצוות הקטעים (לכל a,b , מתקיים $a\neq x$, $b\neq x$).

סבוכיות INSERT	
סבוכיות DELETE	
סבוכיות CUT(x)	

תאור מבנה הנתונים :

תאור insert:

תאור CUT:

<u>שאלה 4 (18 נק')</u>

נגדיר משקל של צומת בעץ חיפוש בינרי: 1 אם הוא עלה, 2 אם יש לו בן יחיד ו-3 אם יש לו שני בנים.
יהי משקל העץ סכום משקלי כל הצמתים שלו.
לחישוב משקל העץ אחרי הוספה או מחיקה – מחשבים מחדש את משקל כל אחד מהצמתים, ומסכמים.

א **(9 נק')**. נניח שמשקל עץ T הוא בדיוק W. מה יהיה משקל העץ אחרי מחיקת צומת?

לכל היותר:	לכל הפחות:

הסבר:

ב. **(9 נק')** נניח שבעץ T יש n צמתים. מה משקל העץ?

לכל היותר:	לכל הפחות:

הוכחה:

<u>שאלה5 (18 נק')</u>

רוצים מבנה נתונים שיענה על הדרישות הבאות :

פעולות המבנה :

Insert(K) – מכניס את הערך K למבנה.

Member(K) – עונה True אס"ם K במבנה, אחרת עונה False.

DeleteRange(x,y) – מוחק מהמבנה את כל האיברים K כך ש : x<K<y.

סיבוכיות כל פעולה במבנה צריכה להיות O(lg(n)) Amortize .
(אין חשיבות לסיבוכיות ה- Worst Case של כל אחת מהפעולות בנפרד).

תאור המבנה :

תאור פעולת DeleteRange(x,y) :

ניתוח סיבוכיות Amortize :

מס׳ תלמיד ___________ . מס׳ מחברת _____

קייץ 2000, מועד א׳

פתרון מבחן במבני נתונים

מרצה: גיא קינדלר

מתרגל: עודד שוורץ

משך הבחינה: 3.5 שעות + 0.5 שעות הארכה = 4 שעות (וזה סופי)

הנחיות

- ⇦ לפני תחילת המבחן, רשמו את מס׳ התלמיד ומס׳ המחברת שלכם בראש כל אחד מדפי הבחינה.
- ⇦ יבדקו דפי הבחינה בלבד - המחברות ישמשו לטיוטה ולא יבדקו. כתבו את תשובותיכם במחברות תחילה והתאימו את הניסוח ורמת הפירוט שלהן למסגרת המוקצה עבורן. העתיקו את התשובות למחברת הבחינה רק אחרי ניסוחן.
- ⇦ הקפידו על ניסוח מדויק של תשובותיכם. מריחות יידחו.
- ⇦ קראו כל שאלה **במלואה** לפני פתרונה - תשובה לסעיף אחד יכולה להיות תלוייה בסעיף שאחריו.
- ⇦ הניקוד המופיע בצד השאלות והסעיפים מביע כוונה כללית בלבד. ניקוד הסעיפים הסופי עשוי להשתנות. שימו לב כי סכום הנקודות גדול ממאה.
- ⇦ אינכם מחוייבים לתאר אלגוריתמים בפסאודו-קוד, אלא אם כן אתם מתבקשים לעשות זאת במפורש.
- ⇦ הקפידו על שימוש נכון בסימון O, Θ, ו- Ω.
- ⇦ בכל מקום בו אתם מתבקשים לכתוב סיבוכיות של אלגוריתם שכתבתם, הכוונה לסדר הגודל (Θ) של פונקציית הסיבוכיות.
- ⇦ ההנחיות לעיל אינן מלאות. פרשו את השאלות בתום לב, ופנו למרצה או למתרגל במקרה של ספק בקשר לכוונת כותב השאלה.

מס' תלמיד ____________ מס' מחברת _____

1.(20) עבור צומת v בעץ בינארי T, יהא n(v) הצומת שטיול LDR בעץ יבקר בו מיד אחרי שיבקר ב- v.

א) (11) כתבו אלגוריתם יעיל בשם next(v) אשר בהינתן עץ בינארי T, ומצביע p לצומת v בעץ, מחזיר מצביע ל- n(v) או nil, אם v הוא הצומת האחרון בו מבקר טיול LDR ב- T.

האלגוריתם:

אם ל- v יש בן ימני, נרד אליו, ואז נרד שמאלה בעץ עד שאי אפשר עוד. נחזיר מצביע לצומת שאליו הגענו.

אחרת: נעלה למעלה בעץ, ובכל צומת אליו נגיע נבדוק אם הגענו מבנו השמאלי - כאשר מגיעים לצומת כזה עוצרים ומחזירים מצביע לו. אם מגיעים לשורש מבלי למצוא צומת כזה, מחזירים nil.

סיבוכיות האלגוריתם כפונקציה של גובה העץ h: $\Theta(h)$

סיבוכיות האלגוריתם כפונקציה של המרחק d בין v ל- n(v): $\Theta(d)$

מס' תלמיד ____________ . מס' מחברת _____

ב) (9) נתבונן באלגוריתם הבא לטיול בעץ T, בהינתן מצביע p לשורשו :

```
LDR(p);
While  p.left is not nil
        p<-p.left
While p is not nil
        print(p.key)
        p<-next(p)
```

מהי סיבוכיות האלגוריתם כפונקציה של מספר הצמתים בעץ ? $\boxed{\Theta(n) \quad [n=|T|]}$

הוכיחו את תשובתכם {דרך אפשרית : אינדוקציה} :

נראה באינדוקציה כי עבור קבוע c גדול מספיק, הסיבוכיות קטנה מ- cn, ובכך נוכיח כי הסיבוכיות היא $O(n)$ (ברור כי היא $\Omega(n)$). בסיס האינדוקציה ברור. שלב המעבר:

בריצה על עץ בעל n צמתים ששורשו D ותתי העץ התלויים עליו הם L ו- R, האלגוריתם יורד מ- D לשורש של L (אם L לא ריק), אז הוא מבצע מה שהיה מבצע לו היה נקרא עם השורש של L כפרמטר, אז הוא עולה מהשורש של L חזרה ל- D ומדפיס אותו, יורד לשורש של R, מבצע את מה שהיה מבצע לו היה נקרא עם השורש של R כפרמטר, ושוב עולה מהשורש של R ל- D. מכאן שהזמן הוא זמן הריצה על L ועוד זמן הריצה על L (שביחד חסומים ע"י $c(n-1)$), ועוד מס' קבוע של פעולות. אם c קבוע גדול מספיק, סה"כ הזמן חסום ע"י cn, כפי שיש להוכיח.

מס' תלמיד ____________ מס' מחברת _____

.2 (23)

א) (8) נעיין באלגוריתם הבא, המקבל כקלט שני מספרים שלמים אי-שליליים :

```
calc(x,y);
    If y=0 return 1
    If y is even return calc(x·x,y/2)
        else [y is odd] return x·calc(x·x, (y-1)/2 )
```

האלגוריתם מחשב את : $\boxed{X^Y}$

סיבוכיות האלגוריתם היא (בהנחה שפעולות אריתמטיות מתבצעות בזמן O(1) לפעולה) : $\boxed{\theta(log(Y))}$

הוכחת נכונות לסיבוכיות :

Y מחולק ב- 2 בכל שלב (עד כדי ±½). ולכן תוך log(Y) קריאות רקורסיביות הפונקציה תגיע לתנאי העצירה. בכל אחת מרמות הרקורסיה מבוצע מספר פעולות קבוע.

מס' תלמיד ____________ מס' מחברת _____

ב) (8) יהא A מבנה נתונים, ויהי n מספר הפעולות שבוצעו על A מאז איתחולו. נדון בסיבוכיותן של פעולות על A כפונקצייה של הפרמטר n.

האם יכול להיות שסיבוכיות ה- worst-case של **פעולה ספציפית** על A תהיה **קטנה** מה- amortized complexity של הפעולות על A?

כן

נימוק:

לדוגמה: ניקח מבנה עם $\Omega(log(n))$ amortized complexity לפעולה, ונוסיף פעולה בשם PULP, אשר לא מבצעת כלום וחוזרת מיד. סיבוכיות ה- w.c. של PULP היא $O(1)$, אך סיבוכיות ה- amortized לכל פעולה לא השתנתה.

האם יכול להיות שסיבוכיות ה- worst-case של **כל אחת מן הפעולות** על A תהיה **גדולה** מה- amortized complexity של הפעולות על A?

כן

נימוק:

לדוגמה: מחסנית עם פעולת Push(x) בלבד, שמבצעת פעולה כן כפעיל, למעט המקרה ש-X שלילי ואז המחסנית מרוקנת איבר אחרי איבר.

ברור שסיבוכיות ה- worst-case של הפעולה הוא $\Omega(n)$.

כל איבר נכנס פעם אחת ויוצא פעם אחת, לכן ה- amortized complexity של הפעולה הוא $O(1)$.

מס' תלמיד ____________ **מס' מחברת** _____

ג) (7) תארו מבנה נתונים התומך בפעולות insert(x) ו- delete(x) בזמן O(1) worst-case ו- find(x) בזמן worst-case O(nlog n), כאשר n הוא מס' הפעולות שבוצעו על המבנה מרגע איתחולו. בנוסף, המבנה מבצע כל פעולה ב- amortized-complexity O(log n).

המבנה יהיה עץ חיפוש בינארי מאוזן (למשל AVL), ורשימה מקושרת שמשמשת ל"צבירת" פעולות insert ו- delete.

פעולות אלא לא נבצע אלא "נצבור" על ידי הרשימה.

רק כשתיקרא פעולת חיפוש נבצע את כל הפעולות שברשימה על העץ (ונרוקן את הרשימה) ואז נבצע את החיפוש.

3. (23) במודל ההשוואות, השואת גודלם של שני איברי קלט היא הפעולה היחידה המותרת על הקלט. נרחיב את מודל זה, ונתיר גם לקרוא לפעולה compare7(x,y) עבור איברי קלט x ו-y. פעולה זו מחזירה 1 אם $|x-y|\leq 7$, ו- 0 אחרת.

א) (12) האם במודל המורחב ניתן למיין n איברים בפחות זמן מאשר $\Omega(n\log n)$

לא

נימוק:
אותה הוכחה כמו הוכחת עץ החלטה: בכל שלב תשובת השאילתה (השוואה או compare7) מחלקת את קבוצת הפרמוטציות האפשריות לשניים.

ב) (11) הראו אלגוריתם יעיל במודל המורחב אשר בהינתן סדרה של n איברים, מכריע האם יש בה שני איברים שהפרש קטן או שווה 7.

סיבוכיות האלגוריתם: $O(n\lg n)$

האלגוריתם:
נמיין בעזרת מיון שמבטיח זמן $O(n\lg n)$ (למשל MergeSort).

נעבור על כל שני איברים עוקבים ונפעיל compare7 (עוד $O(n)$).

מס' תלמיד ______________ . מס' מחברת ______

ב) (0) הראו חסם תחתון מקסימלי (מבחינת סדר גודל) לאלגוריתם הפותר את הבעיה בסעיף הקודם.

הכוונה היתה לחסום את סיבוכיות כל האלגוריתמים שפותרים את הבעיה מסעיף ב.

החסם: $\Omega(n\log n)$

הוכחה:
הוכחנו חסם תחתון של $\Omega(n\log n)$ לבעיית ה- element distinctness. ההוכחה נשארת תקיפה גם כאשר מגבילים את איברי הקלט להיות כפולות של 8 (בהוכחה לא התייחסנו כלל לערכי האיברים אלא רק ליחס הסדר ביניהם). עתה נשים לב שאלגוריתם הפותר את הבעיה מסעיף ב', כאשר איברי הקלט הם כפולות של 8, פותר עליהם את בעיית ה- element distinctness, ומכאן שסיבוכיותו היא $\Omega(n\log n)$.

מס׳ תלמיד ____________ . מס׳ מחברת _____

4. (23) א) (6) מיכל חישבה את השכיחויות של a,b,c,d,r במחרוזת "abracadabra", בנתה עץ הופמן מתאים T, ואז רשמה את הקידוד הבינארי ע״פ T של המחרוזת הנ״ל. כמה ביטים בקידוד?

23

ב) (10) מצאו מימוש של heap123, התומך בפעולה insert, ובפעולות הבאות, כאשר n הוא מס׳ האיברים במבנה:

stat1 - המחזירה את סטטיסטי הסדר ה- $\lfloor \frac{n}{3} \rfloor$

extract1 - המוחקת את סטטיסטי הסדר ה- $\lfloor \frac{n}{3} \rfloor$

stat2 - המחזירה את סטטיסטי הסדר ה- $\lfloor \frac{2n}{3} \rfloor$

extract2 - המוחקת את סטטיסטי הסדר ה- $\lfloor \frac{2n}{3} \rfloor$

סיבוכיות כל פעולה צריכה להיות O(log n).

תיאור כללי של מימוש המבנה:

נשתמש בערימות מקסימום ומינימום:

ערימת מקסימום שבה יאוכסנו שליש האיברים הקטנים ביותר.

ערימת מינימום שבה יאוכסנו שליש האיברים הגדולים ביותר.

ערימת מינימום-מקסימום שבה יאוכסנו שאר האיברים.

כמו כן נחזיק מונה למספר האיברים בכל אחת מהערמות.

פרטו את מימוש insert(x):

נכניס את X לאחת משלוש הערמות: אם הוא קטן מסטטיסטי הסדר ה- $\lfloor \frac{n}{3} \rfloor$ יוכנס X לערימת המקסימום. אם X גדול מסטטיסטי הסדר ה- $\lfloor \frac{2n}{3} \rfloor$ נכניס את X לערימת המינימום. אחרת נכניס את X לערימת המינימום-מקסימום.

נעדכן את המונה של הערימה אליה הוכנס X. עתה נעביר איברים בין ערימות כך שההפרש בין גדלן של ערימות עוקבות הוא אפס או אחד.

ג) (7) הראו כיצד, בהינתן n איברים, ניתן לאתחל את המבנה מסעיף ב׳ בזמן O(n) :

נמצא את את סטטיסטי הסדר ה- $\lfloor \frac{n}{3} \rfloor$ ואת סטטיסטי הסדר ה- $\lfloor \frac{2n}{3} \rfloor$
בזמן O(n).
נבצע Partition לפי שני האיברים הנ"ל לשלושה מערכים
בזמן O(n). נבנה את הערמות המתאימות בסעיף א׳ כל אחת
בזמן O(n).

מס' תלמיד ____________ מס' מחברת _____

5. (23) בכיתה ראינו מימוש של lookup-table, כלומר מבנה התומך בפעולת find בלבד, ע"י perfect hash. במימוש זה ביצוע פעולת find(x) דרש חישוב של שתי פונקציות hash על x. הראו מימוש של lookup-table אשר תומך באיתחול המבנה ובפעולת find באותה סיבוכיות expected כמו perfect hash (בסדר גודל), אלא שביצוע פעולת find יכלול חישוב של פוקציית hash אחת בלבד!
רמז: השתמשו בטבלת hash אחת, שגודלה ריבועי במספר האיברים.

כיצד מאותחל המבנה:

בהינתן n איברים, נבנה מערך H עם $2n^2$ מקומות, התומך באיתחול ואיתחול מחדש בזמן $O(1)$, ונאתחל את תאי H ל-0 (כל זה לוקח $O(1)$ זמן). עתה נגריל פונקציית hash, h, מתוך אוסף אוניברסלי הממפה את התחום ל-$\{1,..,2n^2\}$.
עבור כל איבר x בקלט נשנה את $H[h(x)]$ ל-1. אם $H[h(x)]$ היה 1 לפני השינוי סימן שיש התנגשות נאפס את תאי H, נבחר פונקציה h חדשה, ונחזור על התהליך.

ביצוע פעולת find(x):

נחזיר את $H[h(x)]$.

חישוב הסיבוכיות של אתחול המבנה :

מספיק להוכיח שתוחלת מס' הפעמים שנבחר פונקציה h חסומה ע"י 2. כדי לקבל זאת, מספיק להראות ש- h (שמוגרלת מתוך קבוצה אוניברסלית כפי שמתואר בסעיף א') אינה גורמת להתנגשות בסיכוי חצי לפחות:

הסיכוי של x מסויים להתנגש עם y מסויים הוא $1/(2n^2)$. מכאן שהסיכוי של x מסויים ליצור התנגשות חסום ע"י $(n-1)/(2n^2)$. הסיכוי הכולל להתנגשות הוא לכן פחות מ- $n^2/(2n^2)=1/2$, כפי שרצינו.

תאריך הבחינה:פברואר 00

מספר קורס: 0368.2158

סמסטר ב׳, מועד ב׳, תש״ס
תאריך הבחינה: 17.2.00

מבחן במבני נתונים
פרופ׳ חנוך לוי, דר׳ יוסי מטיאס, מר עודד שוורץ

הוראות כלליות:

1. במבחן 4 שאלות. יש לענות על כולן.
2. יש למלא **בכל דף** את מספר הסטודנט ומספר המחברת.
3. אורך המבחן שלוש שעות וחצי.
4. הבחינה עם חומר פתוח (כתוב / מודפס בלבד. ללא מחשבים/מחשבונים).
5. יש לכתוב התשובות רק על טופס המבחן.
6. ניתן להשתמש באלגוריתמים שנלמדו בכיתה כקופסאות שחורות.
7. יש למצוא אלגוריתמים עם סיבוכיות טובה ככל האפשר. אלגוריתם יחשב יעיל יותר אם סיבוכיות ה- Worst case שלו יעילה יותר מאלגוריתם אחר, או שהיא שווה לאלגוריתם האחר ויעילות ה - average שלו טובה יותר.
8. עליכם להשיג את מירב הנקודות בבחינה. ציון הבחינה יחושב על פי הנקודות שתצברו.

שאלה 1 (30)

בשאלה הבאה נטפל במפתחות שונים *חיוביים* הלקוחים מטווח מספרים גדול.

א. (9) נתונים N מפתחות ברשימה מקושרת. עלייך למצוא את K (K פרמטר) המפתחות ה"אמצעיים" בגודלם. כלומר את המפתחות שבין האיבר ה N/2-K/2+1 בגודלו ועד האיבר ה N/2+K/2 בגודלו כולל. הניחו ש N ו-K זוגיים. תארו בקצרה את האלגוריתם שתשתמשו בו ואת סיבוכיותו.

סבוכיות האלגוריתמים:

תאור האלגוריתם:

ב. (9) כעת עלינו לטפל במבנה נתונים דינאמי שבו הננו מקבלים את המפתחות אחד אחד ועלינו לתמוך בכל הפעולות הבאות ביעילות:
(מדד היעילות הוא סיבוכיות ה-Worst Case של הפעולה ה"כבדה" ביותר מבין 3 הפעולות).

- insert(x) – הכנסת מפתח x למבנה.
- delete(x) - השמטת המפתח x מהמבנה.
- Find(K) – מציאת האיבר ה- K בגודלו (K פרמטר. K יכול להיות O(n)).

דוגמה: עבור האיברים: 2, 4, 5, 7, 8, 9,12,14, find(2) תחזיר 12.

1. מה יהיה מבנה הנתונים שבו תשתמשי? תארי בקצרה.

2. כיצד תתבצע find(K)? תארי ברמה עקרונית ובקצרה! מה הסיבוכיות?

סיבוכיות:

תאור האלגוריתם:

3. מהי סבוכיות הפעולות האחרות?

Insert(x)

delete(x)

ג. (12) כעת עלינו לטפל במבנה נתונים דינאמי שבו הננו מקבלים את המפתחות אחד אחד ועלינו לתמוך בכל הפעולות הבאות ביעילות:

- insert(x) – הכנסת מפתח x למבנה.
- delete(x) - השמטת המפתח x מהמבנה.
- FindSUM(K) – מציאת סכום האיברים הגדולים או שווים לאיבר ה- K בגודלו (K פרמטר).

דוגמה: עבור האיברים: 2, 4, 5, 7, 8, 9,12,14, find(2) תחזיר 26.

1. מה יהיה מבנה הנתונים שבו תשתמשי? תארי בקצרה.

2. כיצד תתבצע findSUM(K) תארי ברמה עקרונית ובקצרה! מה הסיבוכיות?

סיבוכיות:

תאור האלגוריתם:

מס מחברת ______________

שאלה 2 (30)

אנא קראי את כל סעיפי השאלה לפני שתתחילי לענות.
נתונה קבוצה בת n איברים. כל איבר מכיל שלושה שדות (x,y,z) כאשר x, y ו- z מספרים שלמים מתחום גדול. בכל איבר השדה x נשאר קבוע אך השדות y ו- z יכולים להשתנות.
יש לבנות מבני נתונים יעילים ביותר התומכים בכל הפעולות הבאות:

א. (8) בהינתן a מצא איבר (x,y,z) כך ש- x=a בזמן $O(1)$.

סבוכיות הבניה:

תאור מבנה הנתונים:

ב. (8) .בהינתן (b,c) מצא האיברים כך ששדה ה- y שלהם שווה ל- b ובאיברים אלה החלף את ערכו של השדה ה- y ל- c. חשוב שהזמן המצטבר על סדרת פעולות יהיה יעיל ככל האפשר בתוחלת, אך ניתן לאפשר לפעולות בודדות להיות יקרות.

סיבוכיות:

תארי בקצרה את אופן ביצוע הפעולה:

ג. (7) .בהינתן (b,c) מצא האיברים כך ששדה ה- z שלהם שווה ל- b ובאיברים אלה החלף את ערכו של השדה ה- z ל- c. חשוב שהזמן המצטבר על סדרת פעולות יהיה יעיל ככל האפשר בתוחלת, **וגם שכל פעולה** תהיה יעילה.

.

תאור המבנה, הפעולה והסיבוכיות:

ד. (7) הנח כעת שניתן למחוק איברים. הראה כיצד לתמוך ב- (א) ו-(ב) (אין צורך לתמוך ב-(ג)) תוך הקפדה שבכל רגע נתון, אם מספר האיברים **שנשארו** בקבוצה הוא N, אז הזיכרון שבשימוש אינו עולה על gN, כאשר g איזשהו קבוע.

תאור האלגוריתם והסיבוכיות:

שאלה 3 (20)

נתון עץ חיפוש בינרי ללא מצביעים ל-Parent. נגדיר V(T) קבוצת כל צמתי העץ.
לכל זוג X,Y נגדיר פונקצית משקל W:

W(X,Y) = 1 If Y and X are nodes of T and Y is the left child of X
2 If Y and X are nodes of T and Y is the right child of X
0 Otherwise

לדוגמה:
W(Root[T],Left[Root[T]])=1
W(Left[Root[T]], Right[Root[T]])=0
W(Left[Root[T]], Null)=0

נגדיר משקל העץ S(T) :

$$S(T) = \sum_{\forall x,y \in V(T)} W(x,y)$$

א) (10) מהו S(T) עבור עץ בינרי מאוזן ומלא בעל **2^m - 1** צמתים ?

S(T) =

ב) (10) האם הערך S(T) שציינת בסעיף הקודם הוא תנאי מספיק לכך שעץ בעל **2^m - 1** צמתים הוא עץ מאוזן ומלא ?

מחקי את המיותר: כן לא

הסבר:

שאלה 4 (25)

א) (10) נתון מערך A בגודל n. המערך מכיל O(lg(n)) ערכים שונים.
הציעי אלגוריתם למיון המערך במודל ההשוואות בזמן O(nlglg(n)).

הרעיון:

הסבר:

ב) (10) מר X טוען שאפשר למיין את המערך בזמן $O((\lg\lg(n))^2)$ במודל ההשוואות. האם הוא צודק ?

כן / לא (מחקי את המיותר)

הסבר:

ג) (5) **(בונוס)** מר Y טוען שפיתח אלגוריתם למיון המערך הנ"ל (במודל ההשוואות) בזמן $O(n\lg\lg\lg(n))$. הוכיחי שהוא טועה.

הוכחה: