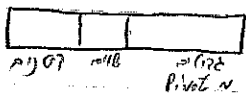


צורה כללית של QuickSort

1. סותרים (שאלה?) איבר כלשהו במערך שיחשב כאיבר זרי "Pivot".



2. מבצעים (מבצעים) Partition מערך ביום של Pivot ומקומים

3. מחנכים באמצעות קריאה רקורסיבית של QuickSort לרצף ימין של המערך וצד שמאל של המערך.

זכרון

1. הזכרון שמימין ה Pivot "במה" במקומו של המערך ואם מקומה סבוכה (מקומה) במקרה הפיך למימי (המסיון - מקומה המיון מעורב).

2. ניתן להפוך את Pivot לאיבר שמימין במערך, למשל בתור [אל] במקרה הקודם ביותר לקיים סבוכות $\Theta(n)$ ניתן להכריז על מערך הממוצע הסוכנות היא קבוע $\Theta(1)$.

3. בוחנים את ה Pivot באיבר מקרה במערך (מחלים) פסס בהסתברות אחת איננו כן $\Theta(1)$ במקרה זה אומר כל קלט = $\Theta(n)$ מסוים לזה שיונים של האזורים הגדולים ביותר $\Theta(n)$ ייתן זמן $\Theta(n)$. הגודל הממוצע $\Theta(n)$ יתן זמן $\Theta(n)$.

תבנית נתונים

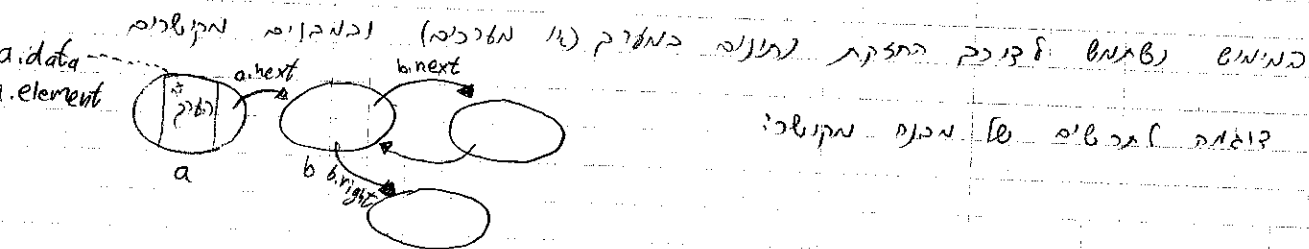
מסקי במחשבים (ADT) Abstracts Data Type (ADT) שצריך לתאר קבוצה של מתיאדות ממשש של תבנית הנתונים מהטיפוס הזה

הפעולה מענישה סוגים

1. יצירה (generation) - יצירה של תבנית נתונים בבסיס מהטיפוס (בד"כ רק תבנית הבדיקה)

2. באזורים עיצובן (update) - לשלש העסקה של נתונים מתבנית הנתונים.

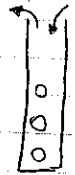
3. שאלות (query) - האם התבנית ריקה? האם התבנית מלאה? מי האבר (בא) המריק? האם כמות התבנית יוב?



(שאלה) שמימין והיא מסוכנת (מקומים) ויעיל (בזמן שאלות) ביום של $\Theta(n)$ מסוכנת הנתונים במערך

מבנה נתונים - מחסנית ומחסנית (הבנה אחת בקוביות)

מחסנית



(Last In First out) תור בקוביות שבו קוביות (ש) באותיות

Make-stack() יצירה (ידידת מחסנית ריקה)

Push(s, x) הכנסה (הוספת x למחסנית s)

Pop(s) היציאה (הוצאת x מהמחסנית s)

empty(s)? האם המחסנית ריקה?

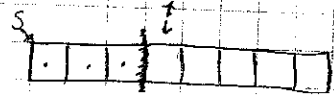
[full(s)?] האם המחסנית מלאה?

Size(s)? כמה איברים קיימים במחסנית?

top(s)? מי (מה) גודל המחסנית?

Make-stack()

create an array S (of length n) $t \leftarrow 0$



empty?(s)

return (t=0)

Full?(s)

return (t=n)

Size(s)

return (t)

top(s)

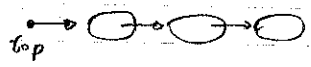
if not empty?(s) return S[t-1] else return error

Pop(s)

if empty?(s) return error
else $t \leftarrow t-1$
return S[t]

Push(s, x)

if full?(s) return error else $S[t] \leftarrow x$
 $t \leftarrow t+1$



מבין המילים המכוננות II

(full? > מלא? >)

Make-stack()

top ← null
size ← 0

empty?(s)

return (top = null)
return (top = 0)

size(s)

return size

top(s)

return top.element

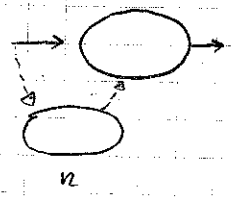
Pop(s)

if empty?(s) then throw error
else temp ← temp.element
top ← top.next
size ← size-1
return temp



Push(s;x)

Create a node n
n.element ← x
n.next ← top
top ← n
size ← size+1



Θ(1) מידת המבנה המכוננות 6 המבנה המכוננות של מ'ב)

III בחזרה למימס ע' מערכים, ולא העבר העקבות, א ג'ים המסונות!

נשנה את המימס של $push$ כך שלם המעק ככה (א), כן גם למעק חכם, ג'ים יוצר, ומתחילים רצף ומתחילים כרצף.

כמה ג'ים כדאי עכבר את המעק החכם?
 נגיד את המעק פ' 2 עם מצב ככה (ולא נמן אחרים במימס המסונות כחוקה עם מעק ג'ים) (א).

זה מוקד כשה מוקדים סיבול' $amortized$ כלומר, בודקים לא מה הסיבול' של בעלת בודק (אלא מה הסיבול' המצטבר של סדרה (מילכה) של פג'ות).

כמה זמן יקח לבצע סדרה של m פעולות (אם המעק כשה מתחילס המסונות לקד במימס סיבול'?

בשלב באת העקבותינו הנו המסונות עם היות m (תיוס)
 עם בעלם העתידים שנתנו להפסד את אום המעק אולי היה $\geq m$, אומר התקדמו אום היות m המעק של המעק.

בידנו הסל' m פעולות, והמסונות עם היות m הכפול של המעק (אכן הסבול' המצטבר של m הכפולות הוא).

$$\theta(m) + \theta(1 + 2 + 4 + \dots + 2^{m-1}) = \theta(m)$$

ב פעולה
 דקות $\theta(1)$
 (א) ה - $push$
 אלא העכבר

הכפולה המסונות העתידים
 הכפולה השניה - 2 העתידים
 השלישית - 4 יכ'ו
 * זה שניה בעק אומר של היות
 היות $m = 2^{m-1}$

* קיבנו הנו שמימס
 באים ממס א פעולה
 אומר לנו $\theta(1)$ אנו
 אחרים שהסיבול'
 של המימס הוא
 ! $\theta(1)$ amortized