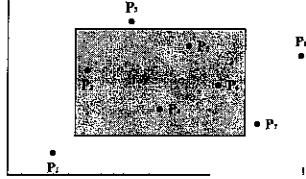


Motivation Example - Reporting (2-D)

Given a set of points S on the plane, preprocess them to build structure that allows efficient queries of the form:

Given an rectangle $R=[x_1,x_2][y_1,y_2]$ find all points in S that are in the rectangle.



הפעלה של אינר הנקודות בתוך המלבן

קריאה (שאלות SQL) שמעביר בקשה

ורוחב.

יש שאלה שמופיעה הרבה במבחנים

עקומה גבוהה עברנו אותה לפני

הנקודות המיועלות במלבן.

Summary

קבץ נתון
20-30 נק' על פי
↓ 16 ו 18

	Arrays	Linked List	Bal. Trees
	Simple, fast	Simple	Not-So-simple
	Inflexible	Flexible	Flexible
Add	$O(1)$	$O(1)$	$O(\log n)$
	$O(n)$ inc sort	$O(n)$ - inc sort	
Delete	$O(n)$	$O(1)$ - any	$O(\log n)$
		$O(n)$ - specific	
Find	$O(n)$	$O(n)$	$O(\log n)$
	$O(\log n)$ ← binary search	(no binary search)	

2

The dictionary ADT

- $Insert(x,D)$
- $Delete(x,D)$
- $Find(x,D)$:

Returns a pointer to x if $x \in D$, and a pointer to the successor or predecessor of x if x is not in D

3

בטוח ה- Delete פה סיבה מבלי המוקד

בעדיניות כיוון, ולכן שמוני ממש מקיני

בכך זה Delete קראת לפני

ואחריהם. עדיין כאן בעדיניות עליון

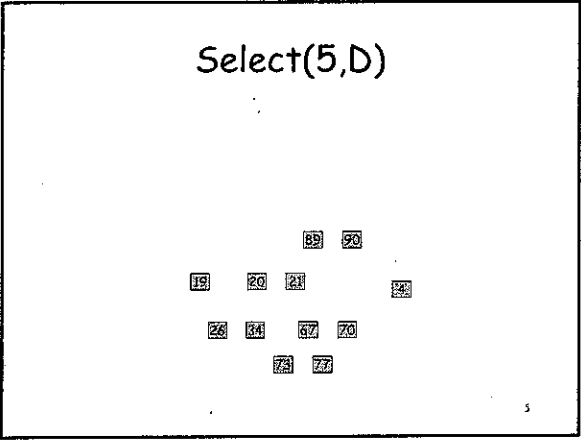
עדיין מקביל.

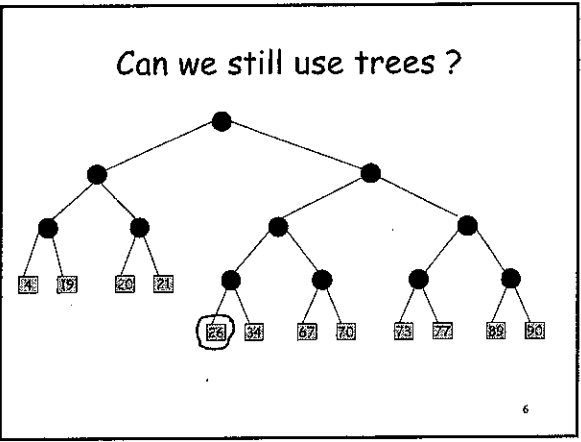
Suppose we want to add to the dictionary ADT

- $\text{Select}(i,D)$: Returns the i^{th} element in the dictionary.

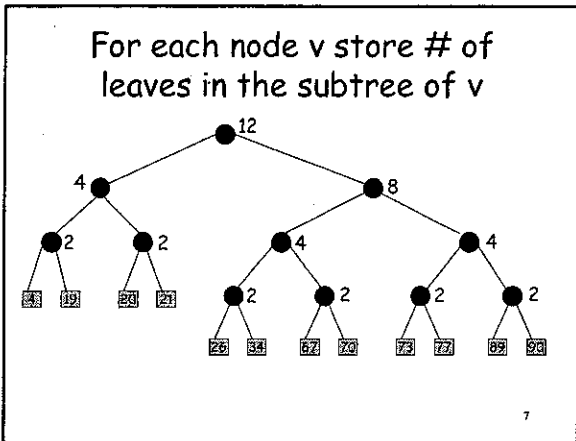
An element x such that $i-1$ elements are smaller than x

כיצד אנו מוצאים את האיבר ה- i במילון?
 בסיוע של $\log n$. אם נבנה רק
 מבנה select בלבד, נצטרך קודם
 לבנות את המלך. כלומר, מילון
 המיון המלא של האיבר i לפני
 יקח $n \log n$. וזה לא טוב.
 (ניתן להימנע מהמיון המלא של האיבר i
 על ידי שימוש ב- $n \log n$).

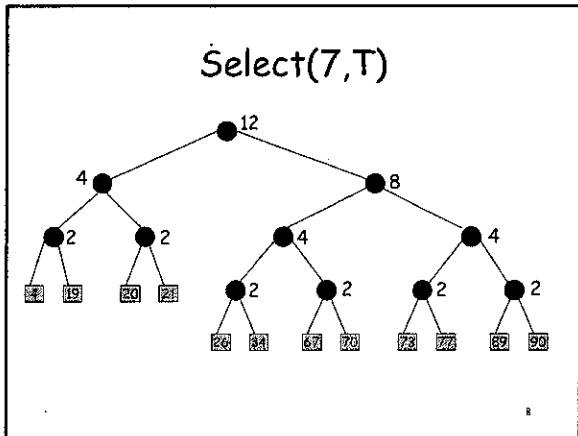




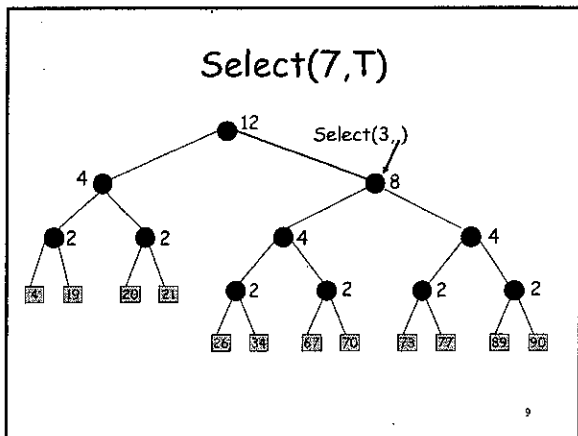
כיצד נוכל להשתמש במילון?
 בעזרת מבנה המילון (במילון) אנו
 יכולים (לעבור) על האיברים insert
 ו- delete יהיה לנו כבוד. אנו
 יכולים על select מילון זה
 לטפל $\log n$ (כל המלך כבר בנוי).
 (יש אף קבצי $\log n$ יותר).



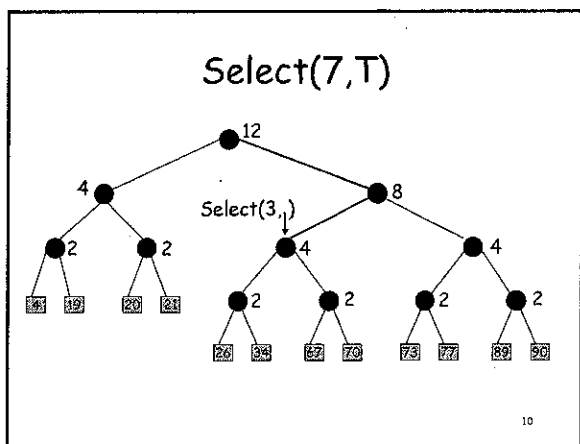
כיצד נעשה זאת?
 (שאלה בלבד) קודם לך את המספר
 הנמנים הנמצאים בתמונה.

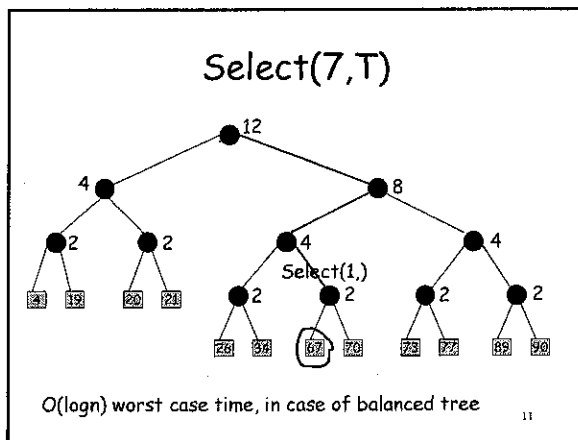


* נרצה לבצע אקסציה באיזה שלב היציאה
 המבוקשת. נסתכל על השלבים ירידה
 שהשלב ה'יציאה' הוא במקום השלם הימני.
 (לפי ימין וכו' שללנו דיוקא ימין)
 נגד הפסגה של העץ ה Select
 כל מה שלם הימני אק הפסגה (מבט
 אל האב) $7 - (8 - 2) = 3$
 ↑ ↑
 ההפסגה הימנית
 הימנית
 הימנית



אם כן, קודם ש האב של השלם הימני
 יהיה נקודת המפגש. כ"ן שללנו שאלה
 לך (מספר) של האב (נקודת המפגש)
 עשנו ממשלם במקום השלם הימני (ממשלם)
 השלם הימני שלו וכן נפחית שלם
 של ההפסגה (נקודת המפגש) אבתי
 את האב של השלם הימני ולכן באמת
 המספר 67



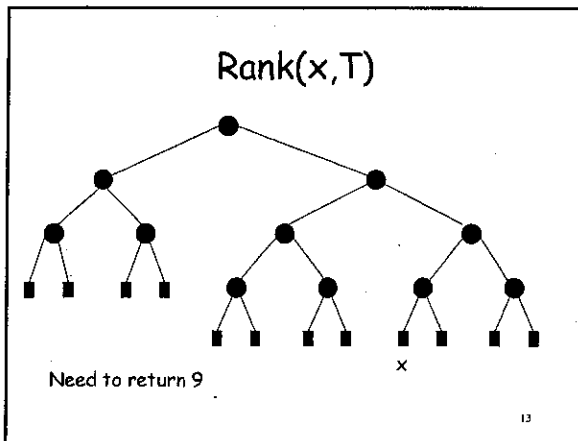


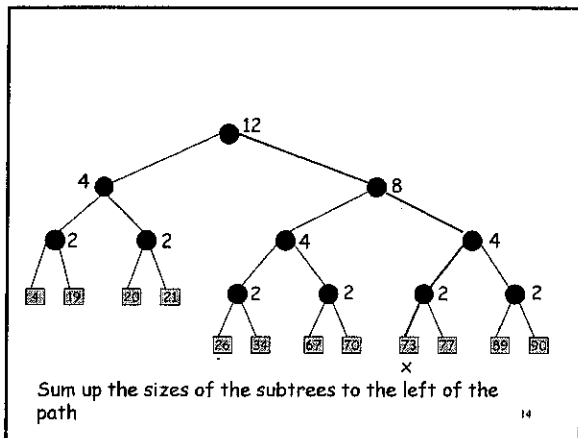
Rank(x,T)

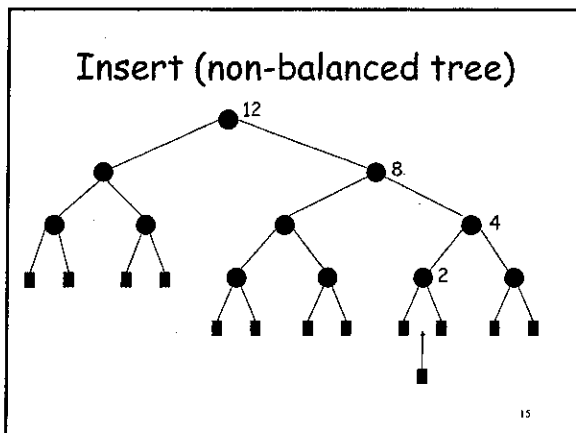
- Return the index of x in T

12

הצגים (אנדרט) של האינדקס של הנתון
 של x ב-T
 כדי למצוא את מספר
 נתון הצגים ש"קטנים או שווים", כלומר
 כל הצגים ש"קטנים או שווים", כלומר
 הצגים הנמצאים במיקומים.
 למצוא את מספר הצגים ש"קטנים או שווים"
 מספרם $4+2+2=8$







Reporting (1-D)

Build a balanced tree over the points
Concatenate them in a list

query: $O(\log n+k)$
space: $O(n)$

הבנת עץ
גמול (אם לא עובד) מילוי העץ

עץ עוקבים את זה (נשמר רשימה)
הקולות היו העלים הם כל צמת ים
אם התחבר הם הם קטן מהם השמאל
העלים נהיו מקושים (ולק מהעלים)
כאן מהם "חיסול הקצק התחתון
של האינטרבל. נשאלים על מה (המיל)
על כן כאן מהם זה ענינה (אם הקצק)
מהמיל העליון.
כמו כל "אוקום - שחור"

Counting (1-D)

Given a set of points S on the line, preprocess them to build structure that allows efficient queries of the form:
Given an interval $I=[x_1, x_2]$ return the # of points in the interval

20

באופן זהו אנו חוצים [קטגוריה כמעט]
אוקום יש מהם (ולא אינפוס ז'אמס)
היה (על מה צ'אמס? הפעם פשוט (ב))
אם המילון Rank(6) - 1 Rank(17)
כיון של התקדמות, כל (בליק של)
Rank עוקמת $\log n$ ומה' סה"כ
המסילות היא $O(\log n)$

Counting (1-D)

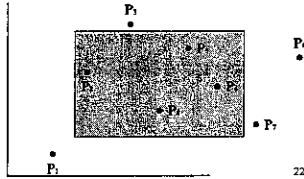
Build a balanced tree over the points, with subtree sizes.
Return: $\text{Rank}(x_2, T) - \text{Rank}(x_1, T) + 1$

query: $O(\log n)$
space: $O(n)$

Reporting (2-D)

Given a set of points S on the plane, preprocess them to build structure that allows efficient queries of the form:

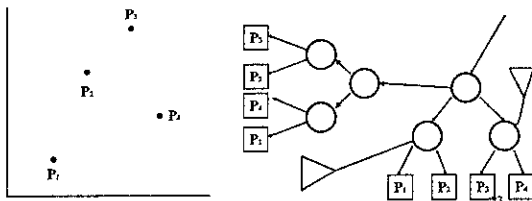
Given a rectangle $R=[x_1, x_2][y_1, y_2]$ find all points in S that are in the rectangle.



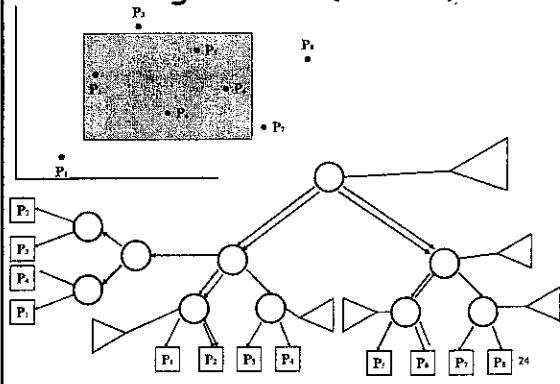
לבנות (למטה ולבן)

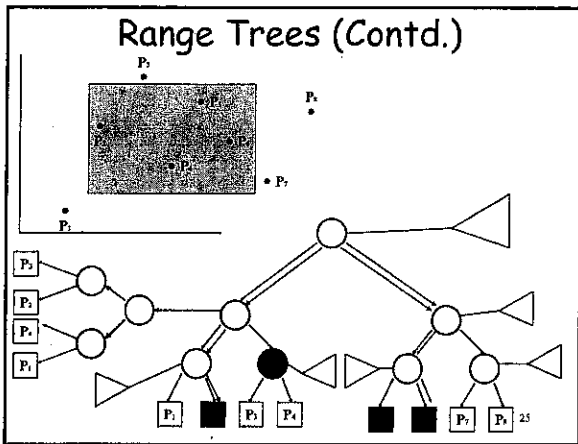
Range Trees (2-D)

Maintain the points in a balanced search tree ordered by x-coor. In each internal node maintain the points in its subtree in a balanced search tree ordered by y



Range Trees (Contd.)





Query processing

- Search by the first dimension gives us $O(\log n)$ trees which together contain the output.
- We search each of these trees to get the answer

26

Analysis (2-D)

- Space $O(n \log n)$
- Query $O(\log^2 n + k)$

27

Further facts

- Generalizes to d-dimensions

28

שאלה 1

תארו מבנה נתונים המתחזק קבוצה של נקודות במישור, כאשר כל נקודה q מיוצגת כזוג מספרים (x, y) .

מבנה הנתונים צריך לתמוך בפעולות $insert(x, y)$, $delete(x, y)$, $find(x, y)$ בזמן גרוע ביותר $O(\log n)$, וכמו כן תומך בפעולה $CountPointsAtDistance(d1, d2)$, המבצעת את הפעולה הבאה:

בהינתן שני מספרים $d1, d2$ מחזירה הפעולה את מספר הנקודות במבנה שמרחקן מהראשית גדול מ $d1$ וקטן מ $d2$ (זיכרו כי מרחק נקודה (x, y) מהראשית הוא).

בסיבוכיות הגמוכה ביותר האפשרית.

29

כמה רדיוס אקטב כמה

x יהי יש בתוך טבלה

ה'א'נ'י. מה רדיוס הקליטה ניתן לחסות

על הקליטה של החישובים קלים

כסדר אקלקולטיבי (המ'נים א'פ' x ו'ט' א'פ'י).

תשובה 1

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

30

פתיחו כדי לחסות את הויסות של חיפוש טבלה.

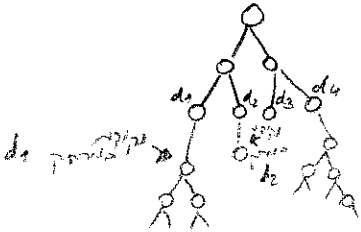
ניכור על שהעץ הראשי שלו יתלה

על המפתחים של הנקודות המתאימות.

ממנה א'פ' מרחק יופים על המ'נים של

ט הנקודות במרחק ה'נ'י. הע'יון המ'ני

יהיה מקסומלי.



סיפור נוסף של העץ הוא זיכור על יחיד של דווח ובו (x, y) והמ'יון יהיה תחילה א'פ' d , ואחרת א'פ' x ואחרת א'פ' y .

האלגוריתם

- מוצאים את הנקודה הרחוקה ביותר מהראשית שמרחקה קטן מ-d2.
- מחשבית את הסדר הסטטיסטי שלה: a.
- מוצאים את הנקודה הקרובה ביותר לראשית שמרחקה גדול מ-d1.
- מחשבית 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.

31

שאלה 2

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 $y=5x$, אם קיימים כאלו (אחרת מחזירה false). הציעו מבנה התומך בפעולות עם סיבוכיות WC הטובה ביותר.

32

תשובה 2

- מבנה הנתונים יהיה עץ חיפוש מאוזן 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.

33

שאלה 3

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 $delete$ | $insert$.

34

תשובה 3

- תיאור האלגוריתם והוכחת נכונות
- לוקחים $n/\log n$ איברים ומכניסים לעץ חפוש מאוזן. עלות כוללת היא $O(n)$. שאר האיברים מוכנסים לרשימה.
- בעת בצוע $find$ מחפשים קודם בעץ ואחר כך ברשימה. עלות חפוש על כל אבר בעץ היא $O(\log n)$. שאר האיברים – $O(n)$.

35

שאלה 4

- נגדיר פרמוטציה מסדר (n, m) באופן הבא:
- מסדרים את המספרים $1, 2, 3, \dots, n$, במעגל, ומצביעים על המספר 1.
- מבצעים m צעדים לאורך המספרים שנותרו על המעגל.
- מדפיסים את המספר אליו הגענו, ומוחקים אותו מהמעגל. אם נותרו מספרים חוזרים לב', ולא עוצרים. מספר שנמחק, אינו נספר במניין הצעדים בשלב ב'.

36

שאלה 4 (המשך)

- לדוגמא: הפרמוטציה מסדר (7,3) היא 4,7,3,1,6,2,5 (ודאו כי אתם מבינים מדוע!)
- תארו אלגוריתם **יעיל** המקבל מספר n , ומחזיר את הפרמוטציה מסדר $(n, n/2)$.
 - תשובה 4: זמן ריצת האלגוריתם $O(n \log n)$

37

תשובה - 4 הסבר

- 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`. נתחיל כשפוינטר מצביע על האיבר 1.
- כשאנחנו באיבר שאינדקסו I , נקפוץ לאיבר שאינדקס שלו הוא $(I+n/2) \bmod k$ בעזרת `OS_Select` כאשר k הוא מספר האיברים שנותרו בעץ (כלומר `size` של השורש).
- נדפיס את האיבר שהגענו אליו, ונמחוק אותו.
- נבצע את 1 ו 2 -עד שלא יישארו איברים בעץ.

