מבני נתונים 08a

תרגול 8

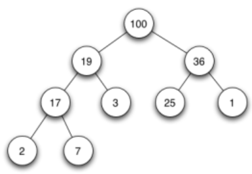
7.2.2008

ערמות

ליאור שפירא

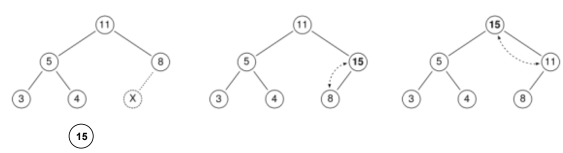
תזכורת: Heaps

- עץ בינארי מלא
- החוק הבסיסי
  - אם צומת B צאצא של צומת A אזי Key(A)≤Key(B)
- הפעולות הנתמכות
  - Find-min
  - Delete-min
  - Decrease-key
  - Insert
  - Merge



תזכורת: Heaps

הוספת צומת (עבור max-heap)



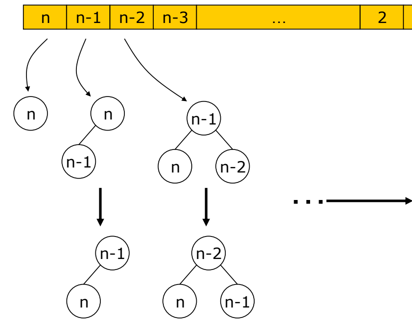
מחיקת השורש

תרגיל 1

- בהינתן מערך באורך n, נרצה ליצור min heap ע"י הכנסה סדרתית של ערכי המערך. הראו סדרת הפעולות לוקחת Ω(nlogn) במקרה הכי גרוע (worst case)
- פתרון
  - נצטרך להראות דוגמה של סדרת ההכנסות שלוקחת Ω(nlogn) פעולות
  - נחפש סדרה ש"תקשה" כמה שיותר על ה-heap

תרגיל 1

n	n-1	n-2	n-3	...	2	1



תרגיל 1

- כל ערך שנוסיף צריך לבעבע לראש העץ
- n/2 ההכנסות האחרונות לוקחות לפחות log(n/2) כל אחת

$$\frac{n}{2}\log(\frac{n}{2}) = \frac{1}{2}n(\log n - \log 2) =$$

$$\frac{1}{2}n\log n - \frac{1}{2}n\log 2 = \theta(n\log n)$$

- מסקנה: W.C = Ω(nlogn)

תרגיל 2

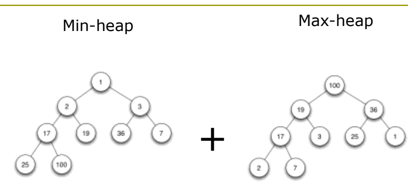
- בהינתן heap שתומך בפעולות extract-min ו- insert בזמן amortized f(n), הראה שניתן למיין מערך מגודל n בזמן O(n·f(n))
- פתרון
  - נבצע n פעולות הכנסה בזמן O(n·f(n))
  - נבצע n פעולות הוצאת מינימום בזמן O(n·f(n))
  - סה"כ O(n·f(n))
- אלגוריתם מיון זה נקרא heap-sort
- בשיעור תלמדו כי מיון n אברים הוא Ω(n·logn)

תרגיל 3 – Median Heap

- ממשו מבנה נתונים התומך בפעולות
  - insert בזמן O(logn)
  - extract-median בזמן O(logn)
  - find-median בזמן O(1)

2	4	5	7	8	12	14	15	20

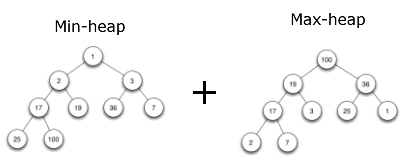
תרגיל 3 - פתרון



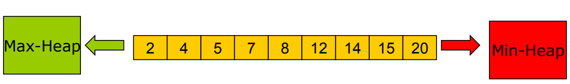
האברים הגדולים (מהחציון)

האברים הקטנים (עד החציון)

תרגיל 3 - פתרון



- נשתמש ב-max-heap ו-min-heap
- n/2 הערכים הגדולים ביותר יישמרו ב-max-heap
- השאר יישמרו ב-min-heap
- החציון תמיד נמצא בשורש של אחד מהם



תרגיל 3 - פתרון

- Find-median
  - If (size(minheap)>size(maxheap))
    - return getmin(minheap)
  - Else
    - return getmax(maxheap)

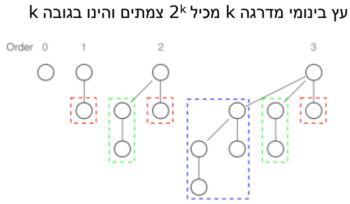
  O(1)
- Insert(x)
  - If (x<getmin(minheap))
    - Insert(maxheap,x)
  - Else
    - Insert(minheap,x)
  - If (abs(size(minheap)-size(maxheap))>1)
    - Balance heaps (move root from bigger heap to smaller heap)

  O(logn)
- Extract-Median
  - Extract median from the max-heap or min-heap…

  O(logn)

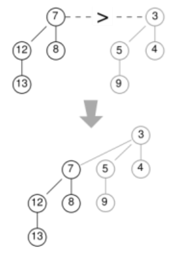
תזכורת: Binomial Heaps

- תחילה נגדיר Binomial Tree:
  - עץ בינומי מדרגה 0 מכיל צומת יחידה
  - עץ בינומי מדרגה k הוא
    - בעל שורש מדרגה k
    - ילדיו הינם עצים בינומיים מדרגות 0,...,k-2,k-1 (בסדר זה)
    - עץ בינומי מדרגה k מכיל 2^k צמתים והינו בגובה k



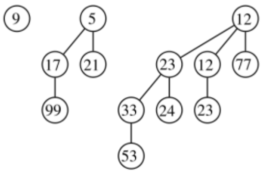
תזכורת: Binomial Heaps

- איחוד שני עצים בינומיים מסדר k-1
  - ניצור עץ בינומי מסדר k ע"י תליית אחד העצים בכך שהמשמאלי ביותר של העץ השני



תזכורת: Binomial Heaps

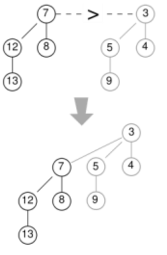
- הגדרת Binomial Heap
  - סט עצים בינומיים המקיימים את התכונות הבאות
    - כל עץ מקיים את תכונת minimum-heap (כל צאצא גדול מההורה שלו)
    - עבור כל סדר k של עץ בינומי, יש 0 או 1 עצים כאלו ב-heap



תזכורת: Binomial Heaps

- פעולת Merge של שני עצים מדרגה k

```
function mergeTree(p, q)
    if p.root <= q.root
        return p.addSubTree(q)
    else
        return q.addSubTree(p)
end
```



תזכורת: Binomial Heaps

- פעולת merge של שני Binomial Heaps

```
function merge(p, q)
  while not( p.end() and q.end() )
        tree = mergeTree(p.currentTree(), q.currentTree())
        if not heap.currentTree().empty()
          tree = mergeTree(tree, heap.currentTree())
          heap.addTree(tree)
        else
          heap.addTree(tree)
              end if
        heap.next() p.next() q.next()
    end while
end
```

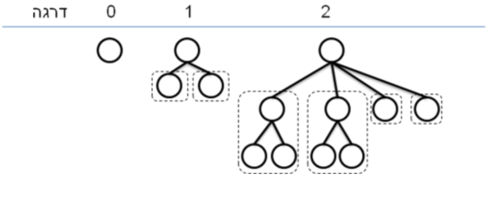
תזכורת: Binomial Heaps

- פעולת Insert
  - ניצור heap חדש המכיל את האבר החדש, ונבצע merge בין שני ה-heaps
- פעולת minimum
  - עלינו לחפש את הערך המינימלי מבין שורשי העצים בheap
- פעולת delete-min
  - מצא את האבר ומחק אותו
  - הפוך את בניו ל-binomial heap ומזג את שני ה-heaps
- פעולת Decrease-min
  - בדומה לפעולות ב-heap רגיל
- פעולת Delete
  - שנה את הערך ל-∞- ובצע delete-min

הדגמה באנימציה של Binomial Heaps: http://www.cse.yorku.ca/~aaw/Sotirios/BinomialHeap.html

תרגיל 4: עצים בינומיים שמנים

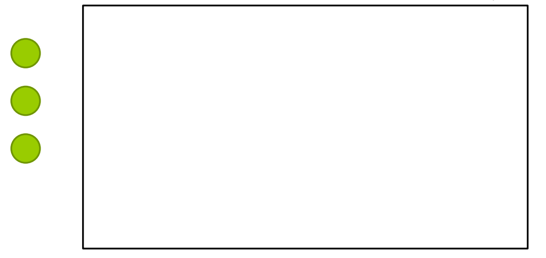
- נגדיר עצים בינומיים שמנים כך:
  - עץ בינומי "שמן" מדרגה 0 מכיל צומת אחד בלבד.
  - עץ בינומי "שמן" מדרגה k ניתן לבנות משלושה עצים בינומיים "שמנים" מדרגה k-1 כאשר נתלה שניים מהם על העץ השלישי.



תרגיל 4: עצים בינומיים שמנים

- כיצד נגדיר ערמה בינומית "שמנה"?

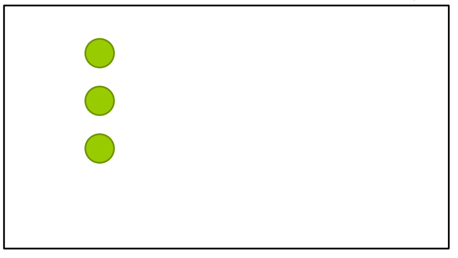
ערמה בינומית שמנה



תרגיל 4: עצים בינומיים שמנים

- כיצד נגדיר ערמה בינומית "שמנה"?

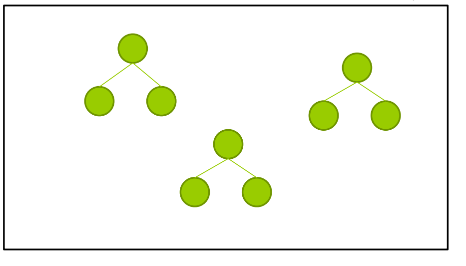
ערמה בינומית שמנה



תרגיל 4: עצים בינומיים שמנים

- כיצד נגדיר ערמה בינומית "שמנה"?

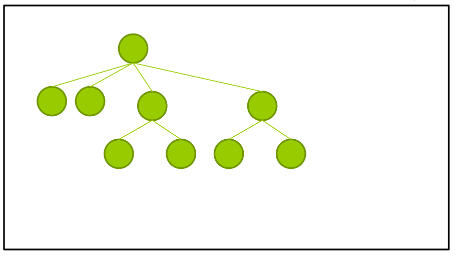
ערמה בינומית שמנה



תרגיל 4: עצים בינומיים שמנים

- כיצד נגדיר ערמה בינומית "שמנה"?

ערמה בינומית שמנה



כיצד תתבצע פעולת meld? למה הדבר אנלוגי?

תרגיל 5(*) לבית

- בהינתן מימוש של Fibonacci heaps בו לא מתבצע cascading cuts, הראו שעבור סדרת m פעולות על מקס' n אברים, עלות פעולה ממוצעת גבוהה ככל האפשר
- קצת תזכורת
  - פעולות decrease-key מאד מהירות (זמן קבוע ממוצע)
  - בעת פעולת extract-min נבצע consolidate
  - דרגת כל צומת הינה D(n) והיא חסומה ע"י O(logn)

הסוף