# מבני נתונים ואלגוריתמים

27 בנובמבר 2011

## תרגיל

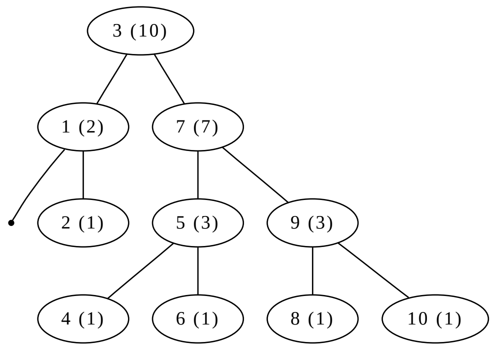
הציגו מבנה נתונים התומך ב:

1. הכנסה
2. הוצאה
3. חיפוש
4. גישה לאיבר ה$k$ בגדלו.

ב$O(\log n)$ כאשר $n$ מס' האיברים במבנה.

### פתרון

ניקח עץ חיפוש שאנחנו מכירים ובכל צומת נשמור גם כמה צאצאים יש לאותו צומת (כולל הצומת עצמו):




נסמן:
אם $N$ = צומת, אז:

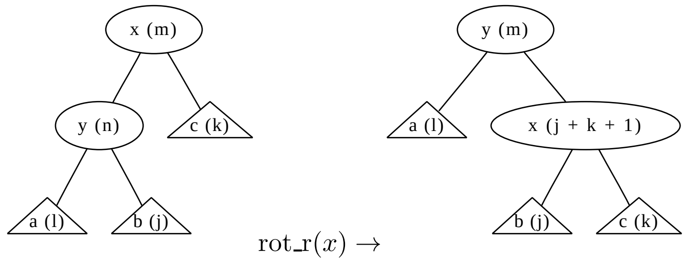
- $\text{Left}(N)$ בן שמאלי של $N$
- $\text{Right}(N)$ - הבן הימני של $N$
- $\text{Count}(N)$ - מס' הצאצאים של $N$ (כולל עצמו)
- $\text{Count}(Null) = 0$

האלגוריתם:

**אלגוריתם 1** גישה לאיבר ה$k$

```
function get_k(T, k):
 left_cnt = Count(Left (T))
 if k == left_cnt + 1
  return T
 else if k <= left_cnt
  return get_k(Left (T) , k)
 else
  return get_k(Right (T) , k − left_cnt − 1)
```

נסביר איך לעדכן את $\text{Count}\left(T\right)$ כאשר מבצעים פעולות של עץ $AVL$.
בהכנסה\הוצאה:
כאשר מכניסים\מסירים צומת, נעדכן את Count של כל האבות הקדמונים של הצומת. כאשר נעשה סיבוב נעדכן את ה-Count באופן הבא:




$\text{rot\_r}(x) \rightarrow$

האלגוריתם:

---

אלגוריתם 2 הכנסה\הוצאה

---

1. מכניסים\מסירים עלה
2. מעדכנים את כל האבות של העלה (רק את Count)
3. חוזרים לעלה ומבצעים תיקון של עץ $AVL$ כרגיל (עם שינוי Count)

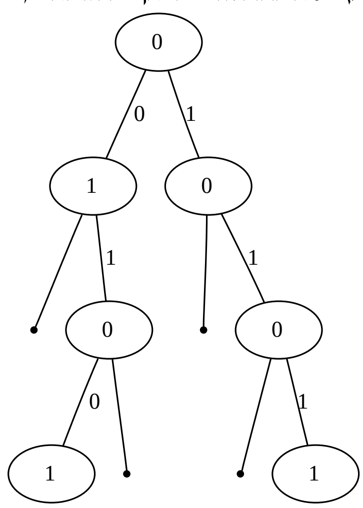
---

# תרגיל

הציעו מבנה נתונים התומך בהכנסה, הסרה וחיפוש של מחרוזות ביטים: 01100...1.
הפעולות הן ב$O\left(k\right)$ כאשר $k$ אורך המחרוזת. (המבנה מתעלם מכפילויות, אין תלות במס' האיברים במבנה).

## פתרון

נשתמש בעץ בינארי. כל צומת יחזיק שני בנים (Left, Right) וביט אינפורמציה (valid).
כל מחרוזת תהיה מסלול בעץ (0 יאמר שמאל, 1 יאמר ימין).
אם מסיימים בצומת בו valid=1 אז המחרוזת במבנה.
עץ שהמחרוזות בו הן לדוגמה 0, 01, 111 יהיה:




האלגוריתמים:

**אלגוריתם 3 הכנסה של מחרוזת**

```
insert(s, n, T):
 if n == 0
  valid(T)=1;
  return;
 else if s[0]==0
  next=Left(T)
  if next=null
   Left(T)=new node
  insert(s+1, n−1, Left(T))
 else //s[0] == 1
  Next = Right(T)
  if Next == null
   Right(T)=new node;
  insert(s+1, n−1, Right (T))
return;
```

סיבוכיות עבור מחרוזת באורך $k$ היא:

$$T_{insert}\left(k\right)=O\left(1\right)+T_{insert}\left(k-1\right)=O\left(k\right)$$

חיפוש והוצאה - תרגיל לבית.

# מיונים

# תרגיל

נתונה רשימה של זוגות:

$$\left(a_1,\, b_1\right),\, ...,\, \left(a_n,\, b_n\right)$$

כאשר $a_n$ ו$b_n$ מציינים תאריך לידה ופטירה של לטאה.
רוצים לדעת מה המס' המקסימלי של לטאות שחיו בו זמנית.

## פתרון

הרעיון: נמיין את רשימת המספרים:

$$a_1,\, b_1,\, a_2,\, b_2$$

כמעט.
פורמלית - לכל זוג $a_i,\, b_i$ נגדיר זוג מספרים:

$$\left(a_i,\, 1\right),\, \left(b_i,\, -1\right)$$

קיבלנו רשימה של $2n$ זוגות $\{(x_i,\, y_i)\}$.
נמיין לפי הקואורדינטה הראשונה - בעלות של $O\left(2n\log\left(2n\right)\right)=O\left(n\log n\right)$.
כעת נעבור על הרשימה הממויינת באופן הבא:

**אלגוריתם 4 מעבר על הרשימה הממוינת לפתרון התרגיל**

```
max = 0
count = 0
for (x_i, y_i) in sorted_list
 count += y_i;
 max = max(max, count)
return max
```

## תרגיל

בהנחות של התרגיל הקודם הציעו מבנה נתונים שמאפשר הכנסת\הוצאה של לטאה וחישוב כמה לטאות חיו בזמן מסוים ב$O(\log n)$ כאשר $n$ = מס' הלטאות.

### פתרון

נשתמש בעץ שבו שומרים את מס' הצאצאים של כל צומת מתרגיל קודם.
נחזיק שני עצים כאלה:
$T_{birth}$ - עץ לידות
$T_{death}$ - עץ פטירות.
הכנסה של $(a_i, b_i)$ תהיה:
$T_{birth}.\text{Insert}(a_i)$
$T_{death}.\text{Insert}(b_i)$
וכנ"ל להוצאה.
בהינתן זמן מסוים $t$, נחפש אותו בכל אחד מהעצים. היות ואנו שומרים כמה צאצאים יש לכל צומת, אז תוך כדי החיפוש אפשר להבין את מיקומו של $t$ ברשימת ה$a_i$ים ורשימת ה$b_i$.
נניח שמתקיים:

$$\begin{aligned} a_r^{'} &\leq t < & a_{r+1}^{'} \\ b_s^{'} &\leq t < & b_{s+1}^{'} \end{aligned}$$

(כאשר $r, s$ המיקום של $t$ בעצים) נחזיר $r - s$.
($a^{'}$ ו $b^{'}$ הן הרשימות הממוינות).

## תרגיל

תהי $L$ רשימת מספרים ממשיים באורך $n$. נניח שאיברי $L$ מתפלגים באופן אחיד בקטע $[0, M]$, מיינו את $L$ ב$O(n)$ (בממוצע).

### פתרון

וריאציה על מיון דליים.

---

**אלגוריתם 5** פתרון באמצעות מיון דליים

---

1. נאתחל $n$ דליים ריקים (דלי = רשימה מקושרת) $B_1, ..., B_n$.
2. לכל $x \in L$:

$$x \to B_{\lceil \frac{x}{M} \cdot n \rceil}$$

3. נמיין כל אחד מהדליים ע"י Insertion sort
4. נשרשר את הדליים וסיימנו.

---

### סיבוכיות

נסמן ב$|B_i|$ את גודל הדלי $B_i$ אז הסיבוכיות היא:

$$O(n) + \sum_{i=1}^{n} \frac{|B_i|^2}{2}$$

הגודל של $B_i$ מתפלג בינומית, וזה קרוב מספיק להתפלגות פואסונית:

$$|B_i| \sim B\left(n, \frac{1}{n}\right) \sim \text{Poi}(1)$$

אז

$$|B_i|^2 \sim B\left(n,\, \frac{1}{n}\right)^2 \sim \text{Poi}\left(1\right)^2$$

מסקנה:

$$E\left(|B_i|^2\right) \approx E\left(\text{Poi}\left(1\right)^2\right) = \Theta\left(1\right)$$

לכן:

$$E\left(\sum_{i=1}^{n} \frac{|B_i|^2}{2}\right) = \sum_{i=1}^{n} \frac{E\left(|B_i|^2\right)}{2} = \Theta\left(n\right)$$

לכן סה"כ העלות היא $\Theta\left(n\right)$.

## תרגיל ( אלגוריתם Select - גרסה הסתברותית)

נתונה רשמה באורך $n$. רוצים את האיבר ה$k$ בגודלו. מצאו אותו ב$O\left(n\right)$.

### פתרון

---

אלגוריתם 6 פתרון - וריאציה על Quick Sort

---

נתחיל כמו בQuick Sort - בוחרים איבר אקראי ומפצלים את הרשימה לפי אותו איבר.
במקום לבצע 2 קריאות רקורסיביות, נבצע:

if length($L_1$)+1==k
 return x
else if length($L_1$) $\geq k$
 return Select($L_1$, $k$)
else
 return Select($L_2$, $k - \text{length}\left(L_1\right) - 1$)

---