1 (21/10/12)

יחסים בין פונקציות

jeniiii.bee@gmail.com ג'נפי

הגדרה

נסמן $f(n)=o(g(n))$ (ט קטנה) אם $\lim_{n\to\infty}\left|\frac{f(n)}{g(n)}\right|=0$

יחסים בין פונקציות

טרנזיטיביות - אם $f(n)=\Theta(g(n))$ ו- $g(n)=\Theta(h(n))$ אז $f(n)=\Theta(h(n))$ (כנ"ל לגבי o, O, Ω)

רפלקסיביות - $f(n)=\Theta(f(n))$ (גם עבור Ω, O)

סימטריה - $g(n)=\Theta(f(n))$ אמ"ם $f(n)=\Theta(g(n))$

סימטריה מתחלפת - $f(n)=O(g(n))$ אמ"ם $g(n)=\Omega(f(n))$

תרגיל

תה"נ $g(n)=2n^2+5n+2$, $f(n)=n^2$ צ"ל: $g(n)=\Theta(f(n))$

פתרון, נמצא $c_1,c_2>0$ ו- $n_0\in\mathbb{N}$ כך שלכל n מתקיים

$$c_1 f(n)\le g(n)\le c_2 f(n)$$

ובמקרה הזה ניתן לקחת $c_2=9$, $c_1=1$

טענה

אם מתקיים $0<\lim_{n\to\infty}\frac{f(n)}{g(n)}=c\in\mathbb{R}$ אז $f(n)=\Theta(g(n))$

הוכחה שכן קיים n_0 כך שלכל $n_0<n$ מתקיים $\left|\frac{f(n)}{g(n)}-c\right|<\frac{c}{2}$

ונקבל $\frac{c}{2}g(n)<f(n)<\frac{3c}{2}g(n)$ ומכאן הנדרש.

דוגמה

נקח את הפונקציות הבאות $f(n)=n$; $g(n)=\{2^n \quad 2^n\le n<2^{n+1}$

ונראה $\frac{1}{2}f(n)<g(n)\le f(n)$ כלומר $f(n)=\Theta(g(n))$

ואכן, יהי n עבורו $2^n\le n<2^{n+1}$ ובפרט $n\ge 2^n\ge n\frac{1}{2}$ כנדרש.

עם זאת, נראה שהגבול המבוקש מעלה אינו קיים:

$$\lim_{n\to\infty}\frac{f(2^n)}{g(2^n)}=1 \quad ; \quad \lim_{n\to\infty}\frac{f(2^n-1)}{g(2^n-1)}=2$$

ולכן הגבול לא קיים!

כללים

1) אין משמעות לקבועים, כלומר לכל $c>0$: $f(n)=\Theta(cf(n))$

2) רק הגורם ה"גדול ביותר" קובע.

למשל אם $\alpha<\beta$ אז $n^\alpha \ll n^\beta$ (כלומר $n^\alpha=o(n^\beta)$)

ועל כן למשל עבור $\alpha_1>\alpha_2>\dots>\alpha_r$ נקבל

$$a_1n^{\alpha_1}+a_2n^{\alpha_2}+\dots+a_rn^{\alpha_r}=\Theta(n^{\alpha_1})$$

3) עבור $\alpha,\delta,\gamma>0$ ו- $\beta>1$ נקבל

$$(\log\log n)^\gamma \ll (\log n)^\delta \ll n^\alpha \ll \beta^n \ll n! \ll n^n$$

דוגמה

נוכיח $(\log n)^3=o(n^{0.001})$, כלומר $\lim\limits_{n\to\infty}\frac{(\log n)^3}{n^{0.001}}=0$

נעשה זאת ע"י לופיטל:

$$\lim\frac{(\log n)^3}{n^{0.001}}=\lim\frac{\frac{1}{n}\cdot 3\log^2 n}{0.001\,n^{-0.999}}=\lim\frac{3\log^2 n}{0.001\,n^{0.001}}=$$

$$=\lim\frac{6\cdot 10^9}{n^{0.001}}=0$$

הגדרות

* $f(n)$ תקרא **פולינומיאלית** אם קיים $\alpha>0$ עבורו $f(n)=O(n^\alpha)$
* $f(n)$ תקרא **אקספוננציאלית** אם קיים $\alpha>1$ עבורו $f(n)=\Omega(\alpha^n)$.

דוגמה

נראה $3^n\neq O(2^n)$, ואכן, נניח בשלילה $3^n=O(2^n)$ כלומר

$$3^n\le c\cdot 2^n$$

$$1.5^n\le c$$

וזו כמובן סתירה (כי אמור להתקיים לכל n מספיק גדול).

אריתמטיקה

$$O(f(n))O(g(n))=O(f(n)g(n))\ ;\ O(f(n)^\alpha)=O(f(n))^\alpha$$

$$O(f(n)+g(n))=O(f(n))+O(g(n))$$

כנ"ל עבור Ω,Θ

תרגיל

הראה שמתקיים $f(n)=\sqrt{1}+\sqrt{2}+\dots+\sqrt{n}=\Theta(n\sqrt{n})$

את זה אפשר לעשות ע"י קירוב עם אינטגרל,

אך נבצע לפי הכלים הקודמים

$$f(n)\le n\sqrt{n}=\Theta(n)\cdot\sqrt{\Theta(n)}=\Theta(n\sqrt{n})$$

$$f(n)\ge \underbrace{\sqrt{\tfrac{n}{2}}+\dots+\sqrt{n}}_{\text{מורידים } \frac{n}{2} \text{ איברים ראשונים}} \ge \frac{n}{2}\sqrt{\frac{n}{2}}=\Theta(n)\sqrt{\Theta(n)}=\Theta(n\sqrt{n})$$

ומכאן הנדרש.

תרגיל

הראה שמתקיים $\max\{f(n),g(n)\}=\Theta(f(n)+g(n))$

ואכן, יהי n ונניח בה"כ ש $f(n)\le g(n)$

$$\Rightarrow \max\{f(n),g(n)\}=g(n)\le f(n)+g(n)$$

$$\max\{f(n),g(n)\}=g(n)=2\cdot\frac{1}{2}g(n)\ge\frac{1}{2}(f(n)+g(n))$$

ומכאן הנדרש.

תרגיל

נתבונן על $n\ln n$ בקטע $[1,\infty)$ שהיא עולה ממש.

תהי $f(n)$ הפונק' ההפוכה ל-$g(n)$, צ"ל $f(n)=\Theta\left(\frac{n}{\ln n}\right)$

צ"ל

$$c_1\frac{n}{\ln n}\le f(n)\le c_2\frac{n}{\ln n}$$

נפעיל את g על האגפים

$$c_1\frac{n}{\ln n}\cdot\ln\left(c_1\frac{n}{\ln n}\right)\le n\le c_2\frac{n}{\ln n}\ln\left(c_2\frac{n}{\ln n}\right)$$

$$\left[=c_1\frac{n}{\ln n}(\ln c_1+\ln n-\ln\ln n)=c_1 n\left(\underbrace{\frac{\ln c_1}{\ln n}}_{0}+\frac{\ln n}{\ln n}-\frac{\ln\ln n}{\ln n}\right)=c_1 n(1+o(1))\right.$$

עבור $c_1<1$ אכן נקבל $c_1\frac{n}{\ln n}\cdot\ln\left(c_1\frac{n}{\ln n}\right)\le n$

באותו אופן $c_2\ge 1$

2 (28/10/12)

נוסחאות נסיגה

ישנן שלוש דרכים לפתרון נוסחאות נסיגה: הצבה, איטרציה ומשפט המאסטר.

שיטת ההצבה

ננחש חסם ונוכיח באינדוקציה.

דוגמה

$$T(n)=\begin{cases} T(n-1)+n^2 & n>1 \\ 1 & n=1 \end{cases}$$

ננחש $T(n)=\Theta(n^3)$, וצ"ל שקיימים $c_1,c_2>0$ עבורם

$$c_1n^3\le T(n)\le c_2n^3$$

עבור $n=1$ נקבל $c_1\le 1\le c_2$ וזה נכון עבור $c_1\le 1\le c_2$.

כעת נניח נכונות עבור n' כך ש-$1<n'<n$, ונוכיח עבור n:

$$T(n)=T(n-1)+n^2\underset{\text{הנחה}}{\le} c_2(n-1)^3+n^2=c_2n^3-[(3c_2-1)n^2-3c_2n+c_2]$$

כעת נדרוש $(3c_2-1)-3c_2n+c_2\ge 0$ וזה מתקיים לכל n עבור $c_2=3$.

באותו אופן עבור החסם השני נקבל $c_1=\frac{1}{4}$, וכך הוכחנו את הנדרש.

שיטת האיטרציה

נפתח נוסחה עד לקבלת סכום אותו נחסום.

דוגמה

$$T(n)=\begin{cases} 2T(n-1)+1 & n>1 \\ 1 & n=1 \end{cases}$$

ולכן,

$$T(n)=2T(n-1)+1=2[2T(n-2)+1]+1=$$
$$=4T(n-2)+2+1=\ldots=$$
$$=2^iT(n-i)+2^{i-1}+2^{i-2}+\ldots+1=\ldots=$$
$$=2^{n-1}T(1)+2^{n-2}+\ldots+1=\sum_{j=0}^{n-1}2^j=\frac{1(2^n-1)}{2-1}=2^n-1$$

ומכאן נקבל $T(n)=\Theta(2^n)$

שיטת המאסטר

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

$$T(n)=\begin{cases}\Theta(n^{\log_b a}) & \exists\varepsilon>0:\ f(n)=O(n^{\log_b a-\varepsilon})\\ \Theta(n^{\log_b a}\cdot\log n) & f(n)=\Theta(n^{\log_b a})\\ \Theta(f(n)) & \exists\varepsilon>0:\ f(n)=\Omega(n^{\log_b a+\varepsilon})\end{cases}$$

דוגמה $T(n)=7T\left(\frac{n}{2}\right)+n^2$

מכיון $n^{\log_b a}=n^{\log_2 7}>n^2=n^{\log_2 4}$

שיפולי עבור ערך ε כלשהו $n^{\log_2 7-\varepsilon}>n^2$ לכן

$$T(n)=\Theta(n^{\log_2 7})$$

דוגמה $T(n)=4T\left(\frac{n}{2}\right)+\Theta(n^2)$

מכיון זה המקרה השני לכן

$$T(n)=\Theta(n^{\log_b a}\cdot\log n)=\Theta(n^2\log n)$$

דוגמה $T(n)=4T\left(\frac{n}{5}\right)+\Theta(n\log n)$

נשים לב $n^{\log_b a}=n^{\log_5 4}<n^{\log_5 5}\cdot\log n=n\log n$

נקבל את המקרה השלישי, כלומר

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

דוגמה $T(n)=5T\left(\frac{n}{5}\right)+n\log_5 n$

כיון נצטרך להשוות בין $n^{\log_5 5}=n^1$

ל- $n\log_5 n$, לעומת שני מקרים להשוות בכזה

בכלל לא אחד מתנאי' המשפט לא מתקיים

נעבור לשיטת האיטרציה.

$$T(n)=5T\left(\frac{n}{5}\right)+n\log_5 n=5\left[T\left(\frac{n}{25}\right)+\frac{n}{5}\log_5\frac{n}{5}\right]+n\log_5 n=$$

$$=5^2T\left(\frac{n}{5^2}\right)+2n\log_5 n-n=$$

$$=5^2\left[5T\left(\frac{n}{5^3}\right)+\frac{n}{5^2}\log_5\frac{n}{5^2}\right]+2n\log_5 n-n=$$

$$=5^3T\left(\frac{n}{5^3}\right)+3n\log_5 n-2n-n=$$

$$=5^4T\left(\frac{n}{5^4}\right)+4n\log_5 n-3n-2n-n=$$

$$=5^iT\left(\frac{n}{5^i}\right)+in\log_5 n\underbrace{-(i-1)n-(i-2)n-\ldots}_{\frac{i(i-1)n}{2}=\Theta(i^2n)}=$$

נניח $T(1)=1$, $n=5^i$ ולמצוא נקבל

$$T(n)=5^{\log_5 n}\cdot T\left(\frac{n}{5^{\log_5 n}}\right)+\log_5 n\cdot n\log_5 n-\Theta(n\log_5^2 n)=$$

$$=n+n\log_5^2 n-\Theta(n\log_5^2 n)=\Theta(n\log_5^2 n)$$

שיטת החלפת משתנים

למשל, עבור $T(n)=2T(\sqrt{n})+\log_2 n$, נסמן $m=\log_2 n$

$$S(m):=T(2^m)=2T(2^{m/2})+m$$

$$S(m)=2S\left(\frac{m}{2}\right)+m=\Theta(m\log m)$$

$$\Rightarrow T(n)=\Theta(\log n\cdot\log\log n)$$

③ (4/11/12)

מבני נתונים

תרגיל פתור $T(n)=5T(\frac{n}{5})+n\log_5 n$

$$T(n)=5[T(\frac{n}{5^2})+\frac{n}{5}\log_5(\frac{n}{5})]+n\log_5 n=$$

$$=5[5[5T(\frac{n}{5^3})+\frac{n}{5^2}\log_5(\frac{n}{5^2})]+\frac{n}{5}\log_5(\frac{n}{5})]+n\log_5 n=$$

$$=5^k T(\frac{n}{5^k})+\sum_{i=1}^{k} n\log_5(\frac{n}{5^{i-1}}) \Rightarrow 5^{\log_5 n}T(1)+n\sum_{i=1}^{\log_5 n}\log_5(\frac{n}{5^{i-1}})=$$

$$=n+n\sum_{i=1}^{\log_5 n}(\log_5 n-\log_5(5^{i-1}))\Rightarrow$$

$$=n+n[\log_5^2 n-\frac{\log_5^2 n}{2}]=n+\frac{n\log_5^2 n}{2}=\Theta(n\log_5^2 n)$$

מחסנית עובדת על עקרון LIFO

דוגמא

נתונה לך המחרוזת $S=(\{[\ ]\})()$ ונבדוק באמצעות מחסנית אם הסוגריים מתאימות אחת לשנייה:

```
for all character c of S:
  if c is one of (,[,{
     push(c)
  else if c is one of ),],}:
     if IsEmpty()=true
        error
     if pop() doesn't match c
        error
  if A.IsEmpty()=False
        error
return true;
```

תור עובד על עקרון FIFO

תרגיל נתונות 2 מחסניות, S_1 מלאה ו-S_2 ריקה. נרצה להעביר את S_1 ל-S_2 שיהיו באותו סדר.

ולכן, ניצור מחסנית נוספת T ונקבל:

```
while S1.IsEmpty==False:
    T.Push(S1.pop());
while T.IsEmpty()==false:
    S2.push(T.pop()
```

וקיבלנו אלגוריתם ב- $\Theta(n)$,

תרגיל

יש n אנשים שיושבים במעגל וממוספרים $n,...,1$

בכל סיבוב נוציא את האיש הראשון מהמעגל עד שהוא ריק. כתוב אלגוריתם שמדפיס את האנשים לפי סדר היציאה

ניצור תור

```
create Q        //
for i=1...n
    Q.enque(i)
while Q.IsEmpty()==false
    for j=1 to n-1
        Q.enque(Q.deque())
    print Q.enque()
```

והסיבוכיות היא $\Theta(n\cdot n)$

<u>תור קדימויות</u> אפשר לממש ע"י כמה תורים.

<u>ערימה</u> עבור איבר במקום i (נתחיל לספור מ-1)

בן שמאלי יתקבל ע"י $2i$, בן ימני באינדקס $2i+1$

והאב במקום $\lfloor \frac{i}{2} \rfloor$

תרגיל

נתונות k רשימות ממוינות באורך n. מזג את k הרשימות לרשימה אחת ב-$\Theta(nk\log k)$. ולכן, נגדיר:

H-heap of pairs of values and indexes of lists

L_r = size nk

$i_1=i_2=...=i_k=0$

$i_r=0$

```
for j=1 to k
    H.push(L_j[i_j],j)                      } klog k

For i_r=0 to nk-k-1
    (a,j)=H.pop()
    L_r[i_r]=a
    i_j++
    if i_j<n
        H.push(L_j[i_j],j)                  } Θ(nklog k)
i_r=nk-k
while H.IsEmpty()==false
    L_r[i_r]=H.pop()                        } Θ(nklog k)
    i_r++
```

תרגיל

תחשב את ABC+xD/ (כ postfix) עם מחסנית

נרצה תוצאה (Ax(B+c))/D

```
create S
for i=0 to n-1
    if J[i] is operand:
       ↑
       מערך 22 עם הרכיבים [illegible]
        push(A[i])
    else
        push(pop(),J[i],pop())  // חישוב
return pop()
```

4 (11/11/12)

עצים

עצי חיפוש

יהי x בעץ חיפוש בינארי, ויהי y קודקוד.

אם y בתת העץ השמאלי של x, אזי $y<x$.

אם y בתת העץ הימני של x, אזי $x\le y$.

מעברים על עץ בינארי

PostOrder(x): PostOrder(Left(x)); PostOrder(Right(x)); Visit x;

InOrder(x): InOrder(Left(x)); visit x; InOrder(Right(x));

PreOrder(x): visit x; PreOrder(Left(x)); PreOrder(Right(x));

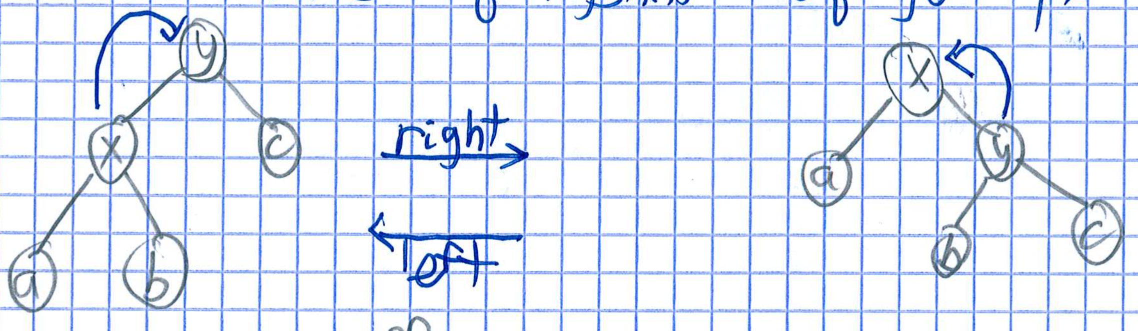
תרגיל

בעץ חיפוש, האיזה סוג מעבר נשתמש כדי שיודפסו לנו המספרים בצורה ממוינת? InOrder.

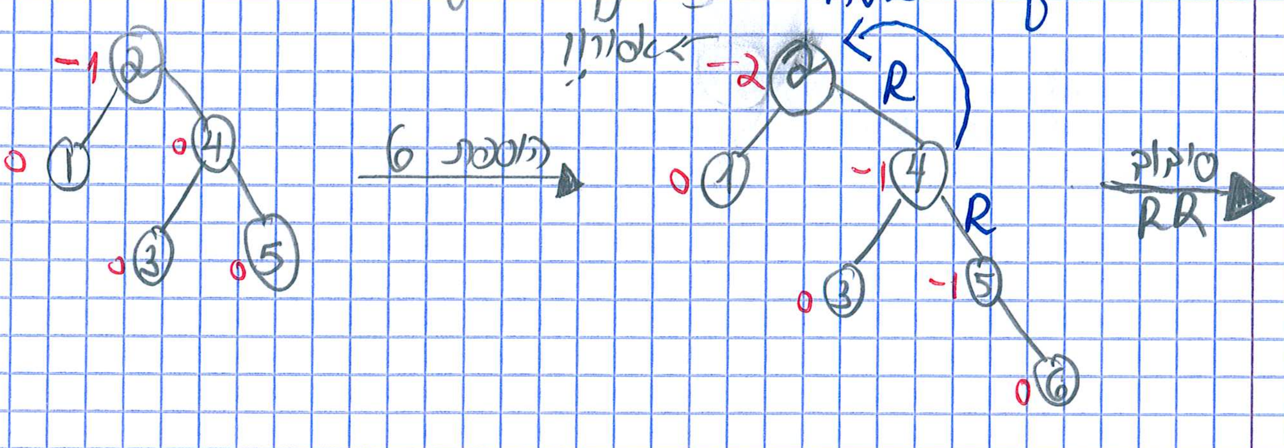
עצים מאוזני גובה $h(T)=O(\log n)$

ההפרש בין גובה תת העץ הימני לשמאלי הוא 1 לכל היותר.

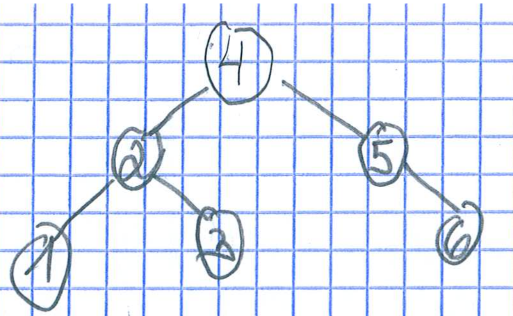
מעבר בין שני עצים מאוזנים ע"י סיבוב:



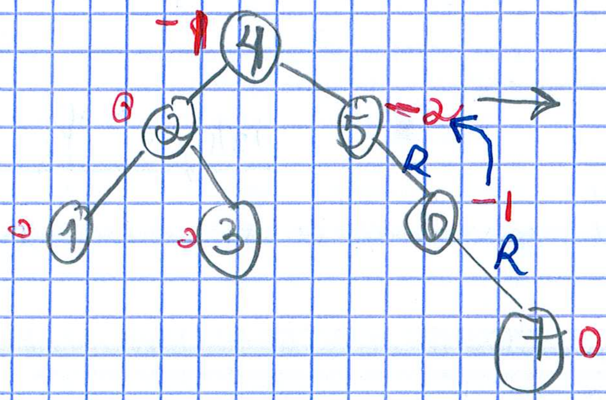
סיבובים בעצי AVL: נתון העץ:



סיבוב
RR

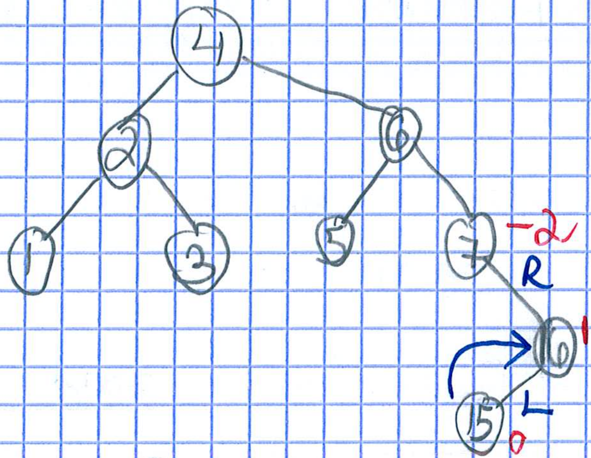


הוספת
7

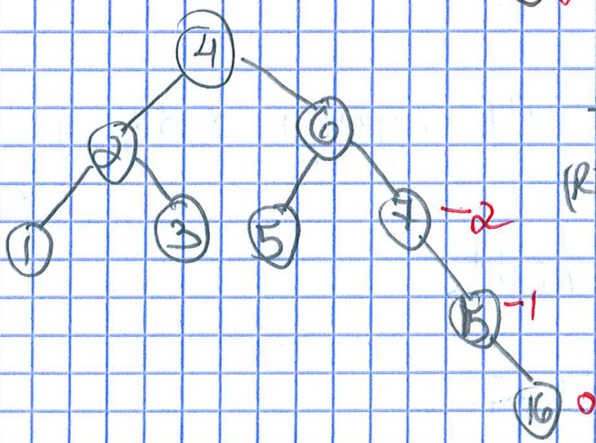


סיבוב

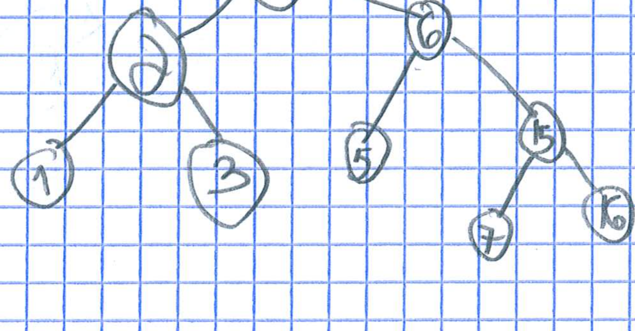
סיבוב RR
והוספת 16
והוספת 15
(לפי הסדר)



סיבוב RL
(שלב ראשון)



סיבוב RL
(שלב שני,
מקביל ל-RR)



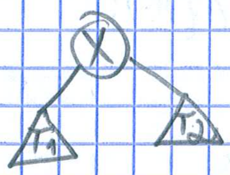
תרגיל

נתונים שני עצי AVL - T_1, T_2 וערך x כך ש- $T_1 < x < T_2$. אחד את T_1, x, T_2 לעץ AVL בצורה יעילה.

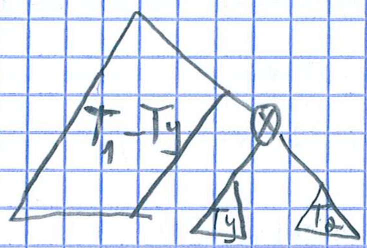
פתרון

נחלק למקרים:

1) $h(T_1)=h(T_2)$, אז נבנה



2) $h(T_1)>h(T_2)$, נחפש בתוך T_1 את תת העץ הימני ביותר שגובהו כגובה של T_2. נסמן שהוא מתחיל בקודקוד y ונסמנו T_y. אז נבנה:



מעבר האלגוריתם יהיה:

א) חפש את תת העץ הימני של T_1 שגובהו $h(T_2)$, שורשו y ונסמנו T_y. (יעילות הריצה $O(\log n_1 - \log n_2)$ כאשר n כמות איברים בעץ).

ב) החלף בין x ל-y.

ג) הוסף את T_y כבן שמאלי של x.

ד) הוסף את T_2 כבן ימני של x.

ה) טפס מ-x לשורש ותקן במידת הצורך.

תרגיל

קוסמטיקאית רוצה לייצג את רשימת הלקוחות שלה לפי מס' תעודה (ת.ז). עבור כל לקוח היא רוצה לציין אם הוא גבר או אישה. הצע מבנה נתונים שתומך בפעולות (ביעילות $O(\log n)$):

Insert(k,c) - הוספת לקוח חדש c עם מפתח k, שהוא אישה.

update(k) - עדכן לקוח k להיות גבר.

FindDiff(k) - מצא ההפרש בין מס' הגברים למס' הנשים בעלי ID נמוך מ-k.

ובכן, נגדיר מבנה X, שמכיל את ה-ID, מין (0/1),
women (מס') נשים תחת העץ שלו
men (מס') גברים תחת העץ שלו,
ואת הלקוח עצמו c.

Insert(k,c):

```
X = new(c)
X.ID = k
X.gender=1
X.women=1
X.men=0          } O(1)
AVL.Insert(X)  //O(log n)
v=parent(X)
while v≠root
    v.women++
    v=parent(v)   } O(log n)
```

Update(k):

```
X = AVL.search(k)    //O(log n)
if X.gender =1;
    X.men=X.men+1
    X.gender=0
    X.women=X.women-1
v=parent(X)
while v≠root     //O(log n)
    v.women -=1
    v.men+=1
    v=parent(v)
```

FindDiff(k):

```
sum-m =0
sum-w=0
T=root
while(T!=null)
    if(T.id<k)
        sum-m+=T.left.men
        sum-w+=T.left.women
        if(T.gender==1)
            sum-w+=1
        else
            sum-m+=1
        T=T.right
    else
        T=T.left
return|sum-w-sum-m|
```

תרגיל

נתונים זוגות מספרים של ממשי, הצע מבנה נתונים שתומך ב:

(א) הכנסה/הוצאה ב- $O(\log n)$

(ב) חיפוש של זוג לפי קואורדינטה ראשונה/שנייה ב-$O(\log n)$.

(ג) החזרה של הזוגות ממוינים לפי הקואורדינטה הראשונה/השנייה ב-$O(n)$.

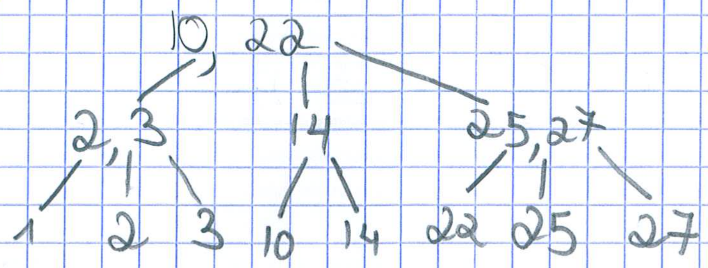
פתרון

נשתמש בשני עצי AVL (או 2-3), שנכניס את הנק' לשניהם - לאחד מהם ממוין לפי הקואורדינטה הראשונה, והשני לפי הקואורדינטה השנייה.

(5) (25/11/12)

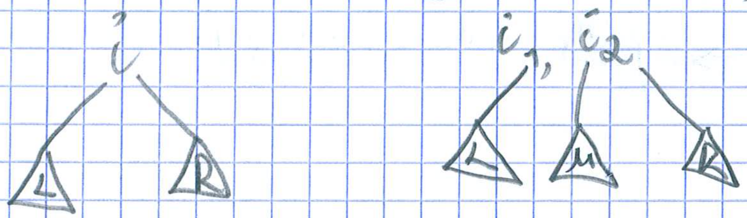
עצי 2-3

כל הערכים בעלים (שכולם באותה רמה) ובשאר הקודקודים יש אינדקסים.



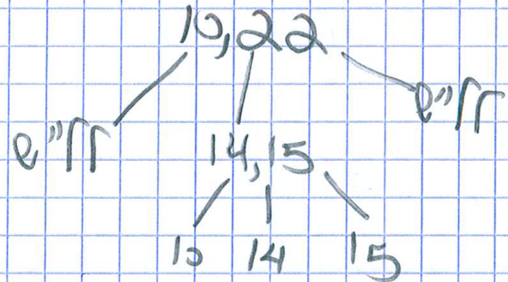
בכל רמה i מתקיים $2^i \le$ כמות הקודקודים $\le 3^i$, ולכן אם h גובה העץ אז $\log_3 L_h \le h \le \log_2 L_h$, כלומר $h=\Theta(\log n)$

ומתקיים

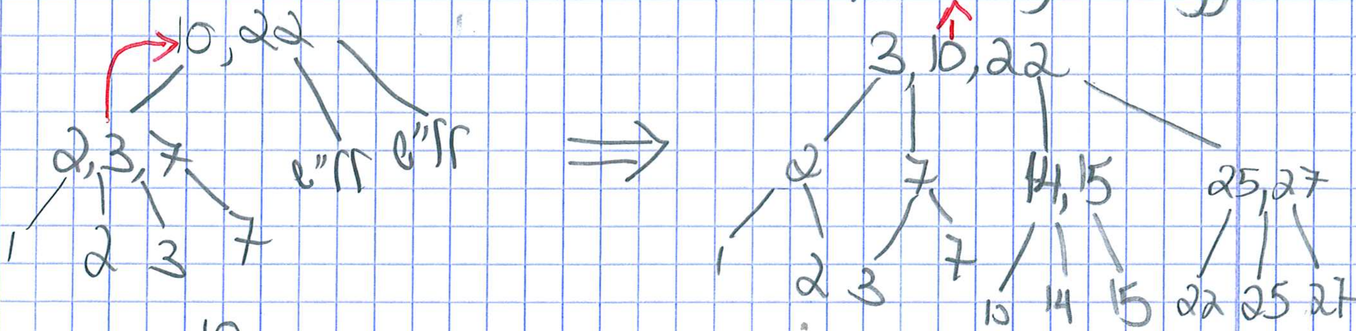


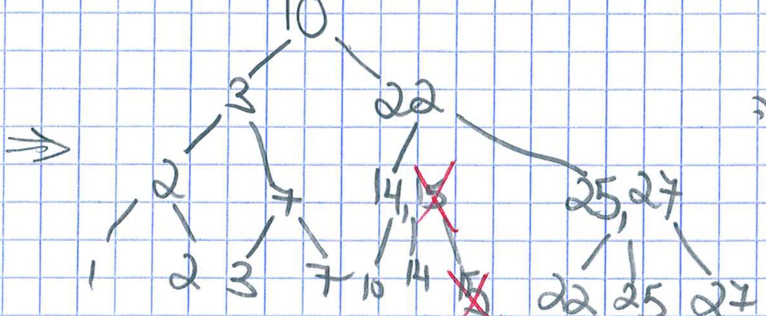
$L < i \le R$ $\qquad$ $L < i_1 \le M < i_2 \le R$

הוספה לעץ: נניח נרצה להוסיף את 15



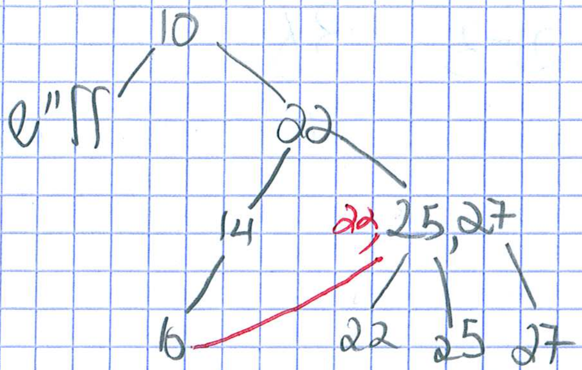
נניח נרצה להוסיף את 7:



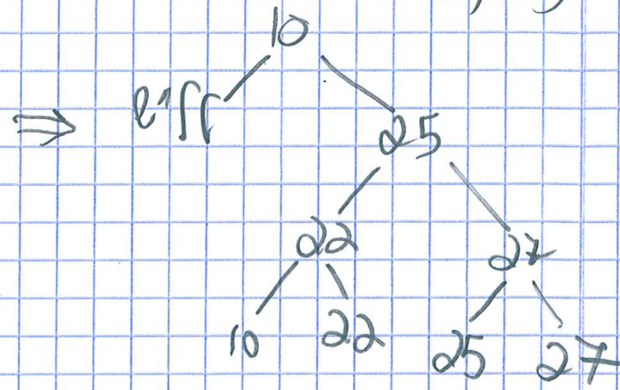


בשחור - העץ לאחר ההוספה

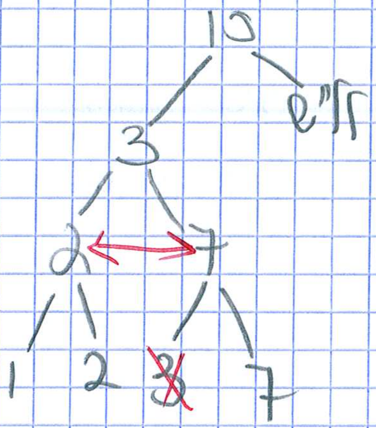
בבורדו - מחיקה של 15



נמחק את 14:



נמחק את 3:



⇒ ממשיכים באותו אופן

תרגיל

הצע מבנה נתונים התומך הכנסה, הסרה וחיפוש של מחרוזת בינ'ים ב- $O(k)$ (k - אורך המחרוזת).

נבנה עץ שמאלה עבור 0 וימינה עבור 1. כמו כן, בקודקודים נשים 0/1 - 0 עבור מקום שאינו סיום מחרוזת ו-1 עבור מקום שיכול להיות סיום מחרוזת.

```
Insert(s,n,T);   // s מחרוזת; n אורך; T מקום בעץ
    if n==0;
        valid(T)=1; return;
    if S[0]==0:
        next=Left[T]
        if next==null:
            Left[T]=new node(0)
        Insert(s+1,n-1,Left[T])
    else:
        next=Right[T]
        if next==null:
            Right[T]=new node(0)
        Insert(s+1,n-1,Right[T]);
```

הרקורסיה היא $T(k)=T(k-1)+O(1)=O(k)$

(5) (25/11/12)

מיונים

תרגיל

נתונה רשימה של זוגות $(a_1,b_1),\dots,(a_n,b_n)$ כאשר a_i לידה של לטאה ו-b_i תאריך פטירה של לטאה, מצאו את המספר המקס' של לטאות שחיו בו זמנית.

מכן, נכתוב כל זוג (a_i,b_i) כשני זוגות $(a_i,1)$; $(b_i,-1)$ ונמיין אותם. אחרי כן נבצע

```
max=0
count=0
for (x_i,y_i) in sorted list:
    count+=y_i
    max=max(max,count)
return max
```

למשל, עבור הרשימה $(1,5),(2,3),(3,10),(4,7),(2,10)$ הרשימה הממוינת תהיה
$(1,1),(2,1),(2,1),(3,-1),(3,1),(4,1),(5,-1),(7,-1),(10,-1),(10,-1)$

i:	1	2	3	4	5	6	7	8	9	10
count:	1	2	3	2	3	4	3	2	1	0
max	1	2	3	3	3	4	4	4	4	4

$\Rightarrow$ יוצא 4

תרגיל

נתונה רשימת מספרים באורך n, ונרצה את האיבר ה-k בגודלו.

מכן, אפשר למיין ואז להוציא את האיבר ה-k (צעד (חנכות) $O(n\log n)$).
כעת נתבונן בפתרון אחר בו המקרה הממוצע הוא $O(n)$.
האלגוריתם שלנו מתבסס על הרעיון של Quick Sort, הוא "יקרא" Random Select ויהיה אלגוריתם הסתברותי.

```
RandomSelect(A,p,r,i): // A - מערך, p - אינדקס התחלתי, r - סוף, i - מיקום רצוי במערך
    if p==r:
        return A[p]
    q=RandomPartition(A,p,r) // חלוקה כלשהי למערך, איבר אקראי
    k=q-p+1 // מיקום בתת המערך
    if i==k:
        return A[q]
    else if i<k:
        return RandomSelect(A,p,q-1,i)
    else
        return RandomSelect(A,q+1,r,i-k)
```

דוגמה: סדר המערך:

(i=5) 7 4 2 1 5 9 8

pivot=4; 2 1 4 7 5 9 8

q=3<i=5

(i=2) 7 5 9 8

pivot=8: 7 5 8 9

p=4 q=6 k=3 i=2

(i=2) 7 5

pivot=7; 5 7 ⇒ ייבחר 7!!

q=5 $\boxed{k=2,\ i=2}$

ניתוח זמן ריצה: במקרה הטוב ביותר (החלוקה באמצע)

$$T(n)=\underbrace{\Theta(n)}_{\text{חלוקה}}+\underbrace{T\left(\frac{n}{2}\right)}_{\text{עבור חצי}}\overset{\text{מאסטר}}{=}\Theta(n)$$

במקרה הגרוע (שה-pivot בקצה ואז מוציאים איבר)

$$T(n)=T(n-1)+\Theta(n)=\Theta(n^2)$$

במקרה הממוצע:

$$T(n)=\frac{1}{n/2+1}\cdot\underbrace{\sum_{i=n/2}^{n-1}T(i)}_{\text{כל האפשרויות לגבולי החלוקה}}+\underbrace{O(n)}_{\text{חלוקה לשתי רשימות}} \le \frac{1}{n/2}\cdot\sum_{i=n/2}^{n-1}T(i)+O(n)$$

אורך תת הרשימה הארוכה ביותר הוא (בקטע) $[\frac{n}{2}, n-1]$, נרצה להראות $T(n)=O(n)$ כלומר שקיים קבוע c עבורו $T(n)\le c\cdot n$.

ולכן, נוכיח באינדוקציה:

$n=1$: $T(1)=O(1) \Rightarrow T(1)\le c$

נניח עבור $n-1$ ונוכיח עבור n:

$$T(n)\le\frac{2}{n}\sum_{k=n/2}^{n-1}T(i)+O(n)=\frac{2}{n}\left(c\cdot\sum_{k=n/2}^{n-1}k\right)+O(n)=$$

$$=\frac{2c}{n}\cdot\left(\sum_{k=1}^{n-1}k-\sum_{k=1}^{n/2-1}k\right)+O(n)=\frac{2c}{n}\left(\frac{n(n-1)}{2}-\frac{\frac{n}{2}\left(\frac{n}{2}-1\right)}{2}\right)+O(n)\le$$

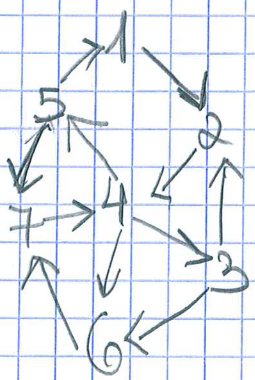
$$\le\frac{3}{4}cn+\underbrace{dn}_{O(n)=}\underset{c\ge 4d}{\le} cn$$

וקיבלנו שבמקרה הממוצע $T(n)=O(n)$ כנדרש.

⑥ (2/12/12)

גרפים ועצים

BFS/DFS



סדר המעבר ב- BFS על הגרף הנ"ל יהיה
1,2,4,5,6,3,7

סדר המעבר ב-DFS:
1,2,4,5,7,6,3

הגדרה - בעיית הסוכן הנוסע (סכנז):

נתון גרף מלא (יש קשתות בין כל זוג קודקודים) וממושקל $G=(V,E,W)$, ונרצה למצוא מסלול שמתחיל ומסתיים באותו קודקוד, עובר בכל שאר הקודקודים פעם אחת ומשקלו המינימלי ביותר האפשרי.

נניח גם ש-G מקיים את אי"ש המשולש:

$$w(v,u)+w(u,k)\geq w(v,k)$$

תרגיל

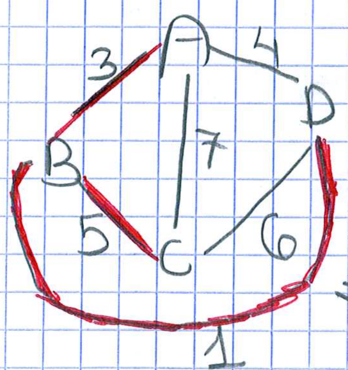
נתון G כנ"ל. יהי $W_{Best-Route}$ המשקל של המסלול הטוב ביותר ב- סכנז. מצאו אלגוריתם פולינומיאלי שמוצא מסלול המעבר בכל צומת בדיוק פעם אחת עם משקל

$$W\leq 2\cdot W_{Best-Route}$$

פתרון

המסלול best-Route בנוי מעץ פורש + קשת נוספת.
עצ"מ - עץ פורש מינ', תמיד יהיה $W_{MST}\leq W_{Best-Route}$
האלגוריתם יהיה:

(1) נמצא MST
(2) נעבור על הגרף ב- DFS (הלוך וחזור).
(3) נשמיט הקודקודים הכפולים.



מסלול עם כפילויות: ⇐ BCBABDB

ובמסלול עם הכפילויות יתקבל

$W \leq 2W_{MST}$

לאחר ההשמטות נקבל $W < 2W_{Best-Route}$

תרגיל

נתון גרף מכוון $G=(V,E)$ בו הקשתות צבועות משלושה צבעים: כחול, אדום וירוק.

הצע אלגוריתם המקבל שני קודקודים u, v ומכריע האם יש מסלול מ-u ל-v כך שכל שתי קשתות סמוכות בצבעים שונים, בסיבוכיות $O(|V|+|E|)$.

פתרון

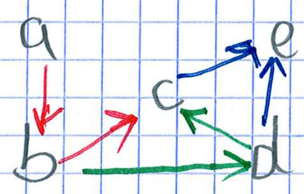
נסמן $C=\{R,G,B\}$

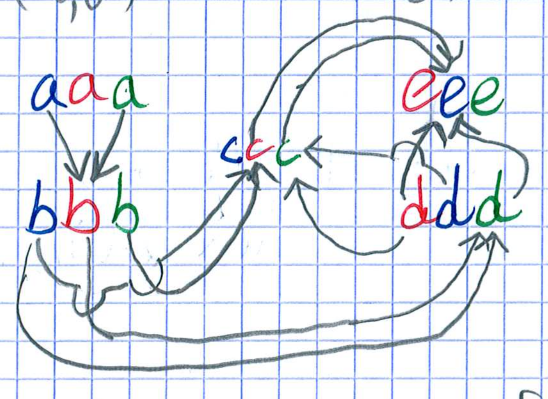
נגדיר גרף חדש G':

V' – מאחד של שלושה עותקים של V: $V'=V_R \cup V_G \cup V_B$

E – לכל $(u,v)\in E$ בצבע c תהיה קשת מ-u_x ל-v_c לכל $c \neq x \in C$.

לדוגמה – אם (u,v) אדומה, ניצור קשתות מ-u_B, u_G ל-v_R.





האלגוריתם יעבוד כך:

1) בנה את G'.

2) עבור $c\in\{R,G\}$:

2.1) נבצע BFS ב-G' מ-u_c. אם הגענו לאחד מ-v_B, v_R, v_G – החזר שקיים מסלול.

3) אחרת החזר שאין מסלול.

(*) בניית G' ו-BFS הם $O(|E|+|V|)$ ולכן בסה"כ הסיבוכיות $O(|E|+|V|)$.

כעת, נוכיח נכונות:

נוכיח ~~לבה~~ שקיים מסלול מ- $\{u_R,u_G\}$ ל- $\{v_R,v_G,v_B\}$

אמ"מ קיים מסלול מתאים ב-G.

(⇐) נניח $u=u^{(0)}\to u^{(1)}\to\dots\to u^{(t)}=v$ מסלול ב-G המקיים

את הנדרש. לכל i נסמן $c_i=color(u^{(i-1)},u^{(i)})$.

לפי הגדרת E' קיימת הקשת מ- $u_{c_i}^{(i)}$ ל- $u_{c_{i+1}}^{(i+1)}$

ב- G' לכל $1\le i\le t$. בנוסף יש הקשת מ- $u_c^{(0)}$

ל- $u_{c_1}^{(1)}$ לכל $c\ne c_1$ ותמיד קיים $c\in\{R,G\}$ כנ"ל.

לכן קיים מסלול ב- G' מ- $\{u_R,u_G\}$ לאחד

מ- $\{v_R,v_B,v_G\}$ הוא

$$u_c=u_c^{(0)}\to u_{c_1}^{(1)}\to\dots\to u_{c_t}^{(t)}=v$$

(⇒) נניח שיש מסלול ב- G' הוא $u_{c_0}\to u_{c_0}^{(0)}\to\dots\to u_{c_t}^{(t)}=v$

מהגדרת E' נובע שלכל $0\le i\le t$ מתקיים

$$color(u^{(i+1)}),(u^{(i)},u^{(i-1)})\in E$$

אז $u=u^{(0)}\to u^{(1)}\to\dots\to u^{(t)}=v$ הוא מסלול ב-G

בו הקשתות הסמוכות הן בצבעים שונים.

תרגיל

יהי $G=(V,E)$ גרף מכוון וקשיר עם פונק' משקל

$w:E\to\mathbb{N}$ בו כל קשת צבועה בשחור או בצהוב.

הצע אלגוריתם המוצא את העפ"מ הצהוב ביותר.

פתרון

נגדיר w חדשה

$$w'(e)=\begin{cases} w(e)-\frac{1}{n} & e \text{ צהובה} \\ w(e) & e \text{ שחורה} \end{cases}$$

האלגוריתם יהיה: (קלט- G,w):

(1) חשב את w' (ב- $O(|E|)$).

(2) נמצא MST עבור w' (ב- $O(|E|\log|V|)$).

כעת, נוכיח נכונות (הצעד השני)

ומכאן, נסמן ב- $y(T)$ את מס' הקשתות הכחולות ב-T
(T - עץ פורש). מתקיים $w'(T) = w(T) - \frac{y(T)}{n}$ $(n=|V|)$.

w תמיד מחזירה ערכים טבעיים, לכן $w(T) \in \mathbb{N}$.

נרצה להראות ש-T הוא העפ"מ הכחול ביותר
אמ"מ T עפ"מ לפי w'.

ממכן, ניקח שני עצים T_1, T_2 עבורם $w(T_1) > w(T_2)$.

$$\Longleftrightarrow w(T_1) - w(T_2) \geq 1$$

כעת, נשים לב שמתקיים:

$$w'(T_1) - w'(T_2) = \left(w(T_1) - \frac{y(T_1)}{n}\right) - \left(w(T_2) - \frac{y(T_2)}{n}\right) =$$
$$= \underbrace{\left(w(T_1) - w(T_2)\right)}_{\geq 1} - \underbrace{\left(\frac{y(T_1)}{n} - \frac{y(T_2)}{n}\right)}_{<1} > 0$$

כלומר קיבלנו $w'(T_1) - w'(T_2) > 0$.

קיבלנו שאם $w(T_1) > w(T_2)$ אז $w'(T_1) > w'(T_2)$.

עבור שני עצים עבורם $w(T_1) = w(T_2)$ נקבל
$w'(T_1) < w'(T_2)$ אמ"מ $y(T_1) > y(T_2)$.

נשלב את שני אלו, ומכאן נקבל את הנדרש.

7 (מומומו)

גרפים וגיבוב

מציאת מסלולים קצרים

1) ממקור יחיד s לכל $v \in V$:

א) משקלים חיוביים בלבד - דייקסטרה $O(E+V\log V)$

ב) משקלים כלליים $O(VE)$

2) פלויד וורשל $O(V^3)$.

רלקסציה (הקלה) של קשת:

הקלה של קשת (u,v) היא הבדיקה האם ניתן לשפר מסלול מ-s עד v דרך הקודקוד u.

תרגיל

נתונים n סוגי מטבעות. שער החליפין בין כל צמד מטבעות נתונים במט' A (מטריצה $n \times n$) כך ש-A_{ij} שער החליפין בין מטבע i למטבע j.
הצע אלג' המכריע האם ק"מת סדרת החלפות המתחילה ומסתיימת באותו מטבע כך שמסיימים עם רווח כספי?

פתרון

נגדיר $G=(V,E)$ עם n צמתים (כל צומת-מטבע) ונשים

$$w_{ij} = -\ln(A_{ij})$$

האלגוריתם יהיה:

1) בנה G כנ"ל.

2) הרץ בלמן-פורד ואם יש מעגל שלילי החזר אותו.

(מ-s כלשהו)

ניתוח סיבוכיות: שלב 1 - $O(n^2)$, שלב 2 - $O(V \cdot E)=O(n^3)$

הוכחת נכונות: אם $v_1, v_2, \dots, v_n, v_1$ מעגל שלילי, אז:

$$-\sum_{k=1}^{n-1} \ln(A_{k,k+1}) - \ln(A_{n,1}) < 0 \iff -\ln(A_{12}A_{23}\cdots A_{n1}) < 0 \iff$$

$$\iff A_{12}A_{23}\cdots A_{n1} > 1$$

תרגיל

הוכח/הפרך הטענות הבאות:

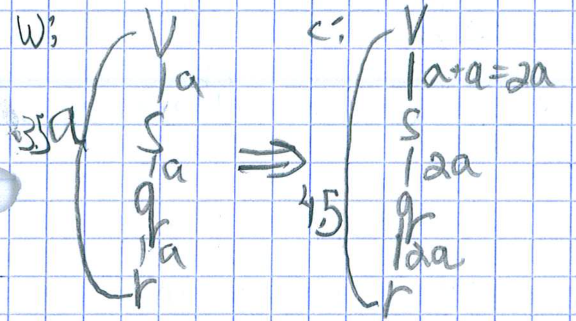
נתון $G=(V,E)$ קשיר לא מכוון עם $W: E \to \mathbb{R}$.

1) נגדיר עבור כל a: $C: E \to \mathbb{R}$ פונק' משקל חדשה כך ש- $c(e)=w(e)+a$. אם p המסלול הקצר ביותר לפי w, הוא גם לפי c.

2) כנ"ל עבור $c(e)=w(e)\cdot a$ $(a>0)$

פתרון

1) לא נכון! דוגמה:



והמסלול בין v ל- r השתנה.

2) נכון, נוכיח:

נניח שקיים מסלול Q עבורו $w(P) > w(Q)$

אז $c(P) > c(Q)$

$$c(Q)=\sum_{e\in Q} a w(e) = \sum_{e\in Q} c(e) < c(P) = \sum_{e\in P} a w(e) = \sum_{e\in P} c(e)$$

$$\Downarrow$$

$w(Q)<w(P)$ בסתירה.

תרגיל

נתון גרף מכוון וקשיר $G=(V,E)$ עם פונק' משקל $W: E \to \mathbb{N}$

הצע אלגו' שמוצא את המסלול הקל הקצר ביותר מ-s לכל $v\in V$.

פתרון

נגדיר $w'(u,v) := |V| w(u,v)+1$

חשיבות גבוהה למשקל על מס' הקשתות

האלגוריתם יהיה:

1) חשב w'

2) הפעל דייקסטרה מ-s לכל $v\in V$ לפי w'.

חישוב סיבוכיות: שלב 1 - $O(E)$, שלב 2 - $O(E+V\log V)$.

סה"כ $O(E+V\log V)$.

הוכחת נכונות:

נרצה להוכיח: P המסלול הקל-קצר לפי w
אמ"ם P המסלול הקל ביותר לפי w'

הוכחה: נקח שני מסלולים פשוטים Q, P כך ש- $w(Q)\ge w(P)$
(פעם אחת הם / הקודקוד שבמסלול / עובר)

$$w'(Q)-w'(P) = \sum_{(u,v)\in Q}(|V|\cdot w(u,v)+1)-\sum_{(u,v)\in P}(|V|w(u,v)+1)=|V|w(Q)+|Q|-|V|w(P)-|P|=$$

$$|V|(w(Q)-w(P))+|Q|-|P| \Rightarrow w'(Q)-w'(P)>0$$

Hash Tables

מבנה נתונים שתומך בחיפוש, הכנסה ומחיקה.
אם יש n נתונים, טבלת הגיבוב תהיה בגודל m כאשר $m<n$.
פונקציה גיבוב h תהיה $h:\{\text{קב' מפתחות}\}\to\{0,1,\dots,m-1\}$
אשר קובעים המקום בטבלה.

התנגשות: $x_1\ne x_2$ אבל $h(x_1)=h(x_2)$

פתרונות:
1) Closed Adressing - שרשור.
2) Open Adressing - rehash.

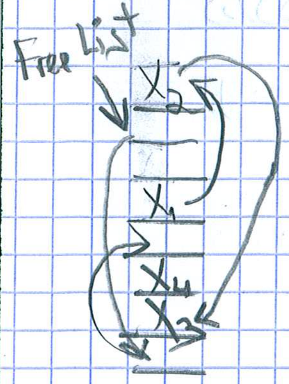
פתרון באמצעות רשימה מקושרת (כל תא בטבלה הוא רשימה):
חיפוש - ברשימה
הכנסה - לראש הרשימה
מחיקה - מוחקים מהרשימה
עבור Closed Adressing

גודל $\alpha=\frac{n}{m}$ הוא האורך הממוצע של הרשימה.
זמן החיפוש יהיה $O(1+\alpha)$ (מניחים שחישוב $h(x)$ הוא $O(1)$).

תרגול

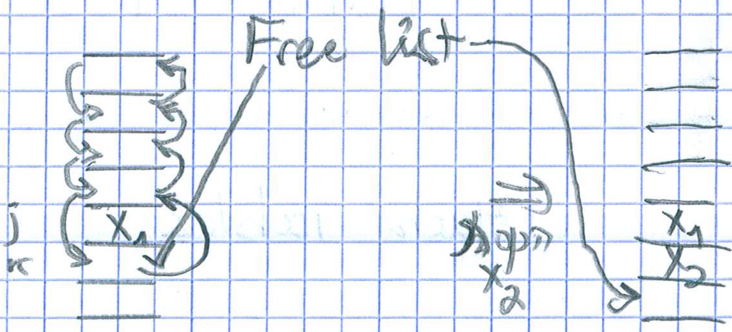
במימוש טבלת גיבוב עם רשימה מקושרת, הצע דרך להקצאה ושחרור של מקומות עבור הערכים בטבלה הזאת.

פתרון



נשתמש ברשימה מקושרת של תאים פנויים,
בטבלה יהיה שדה נוסף - פנוי/לא פנוי:
תא פנוי - שני פוינטרים,
תא תפוס - פוינטר אחד

יהיו $h(x_1)=h(x_2)=j$



$\Rightarrow$ חלה לוגי בצדקים בטלפון

בהוספה: $h(x)=j$

1 מקום j פנוי - מכניסים לתא ומוחקים מרשימת הפנויים.

2 אחרת - בודקים אם $x==T[j]$

(i) אם כן - נשים את x בראש השרשרת הפנויים, עדכון הרשימה ונשרשר לרשימה של j

(ii) אם לא - נשים הערך הישן בתא פנוי, ובמקום הנוכחי את הערך החדש.

מחיקה: אם $h(x)=j$:

1) אם x לא מצביע על שום דבר - מחק והכנס לרשימת הפנויים.

2 אם x משורשר לאחרים - העבר הראשון למקום j ועדכן

אחרת, אם $h(x)\neq j$:

3 מחק מהרשימה והוסף לפנויים.