תרגיל מס.1

עפיף חלומה 302323001

19 במרץ 2009

1 שאלה 1

1. הוכחה עבור O:

(א) צ"ל $f(x)=O(g(x)) \wedge g(x)=O(h(x))$ אזי $f(x)=O(h(x))$

(ב) אם $f(x)=O(g(x))$ אזי קיים $c_1>0$ כך ש שלכל $n_1\in\mathbb{N}, n_1>n_{1_0}$ מתקיים $f(x)\le c_1 g(x)$

(ג) אם $g(x)=O(h(x))$ אזי קיים $c_2>0$ כך ש שלכל $n_2\in\mathbb{N}, n_2>n_{2_0}$ מתקיים $g(x)\le c_2 h(x)$

(ד) נגדיר $n_0=\max(n_1,n_2)$ אזי נובע כי לכל $n>n_0$ מתקיים $f(x)\le c_1 g(x)\le c_1c_2 h(x)$ אזי $f(x)\le c_1c_2h(x)$ כלומר $f(x)=O(h(x))$

2. הוכחה עבור Ω:

(א) צ"ל $f(x)=\Omega(g(x)) \wedge g(x)=\Omega(h(x))$ אזי $f(x)=\Omega(h(x))$

(ב) אם $f(x)=\Omega(g(x))$ אזי קיים $c_1>0$ כך ש שלכל $n_1\in\mathbb{N}, n_1>n_{1_0}$ מתקיים $f(x)\ge c_1 g(x)$

(ג) אם $g(x)=\Omega(h(x))$ אזי קיים $c_2>0$ כך ש שלכל $n_2\in\mathbb{N}, n_2>n_{2_0}$ מתקיים $g(x)\ge c_2 h(x)$

(ד) נגדיר $n_0=\max(n_1,n_2)$ אזי נובע כי לכל $n>n_0$ מתקיים $f(x)\ge c_1 g(x)\ge c_1c_2 h(x)$ אזי $f(x)\le c_1c_2h(x)$ כלומר $f(x)=O(h(x))$

3. הוכחה עבור Θ:

(א) צ"ל $f(x)=\Theta(g(x)) \wedge g(x)=\Theta(h(x))$ אזי $f(x)=\Theta(h(x))$

(ב) $f(x)=\Theta(g(x))$ אזי $f(x)=O(g(x))$ וגם $f(x)=\Omega(g(x))$

(ג) הוכחנו מקודם טרנזיטיביות של Ω וO אזי נובע מזה כי $f(x)=\Omega(h(x))$ וגם $f(x)=O(h(x))$ כלומר $f(x)=\Theta(h(x))$

2 שאלה 2

צ"ל כי $n^2=\Theta\left(4n^2+\log(n)\right)$

מוכיחים כי $n^2 = O\left(4n^2+\log(n)\right)$: צ"ל כי קיים $c>0$ כך שעבורו לכל $n\in\mathbb{F}, n>n_0$ מתקיים $n^2 \le c\left(4n^2+\log(n)\right)$

$$\begin{aligned} n^2 &\le 4cn^2 + c\log(n) \\ 0 &\underbrace{\le}_{c=1} 3n^2+\log(n) \end{aligned}$$

n^2 הוא מספר חיובי ו$\log(n)$ הוא חיובי לכל $n\ge 2$ אזי הביטוי הוא חיובי. מש"ל.

מוכיחים כי $n^2 = \Omega\left(4n^2+\log(n)\right)$ צ"ל כי קיים $c>0$ כך שעבורו לכל $n\in\mathbb{F}, n>n_0$ מתקיים $n^2 \ge c\left(4n^2+\log(n)\right)$

$$\begin{aligned} n^2 &\ge 4cn^2 + c\log(n) \\ 0 &\ge 4cn^2 - n^2 + c\log(n) \\ 0 &\underbrace{\ge}_{c=\frac{1}{8}} \frac{-4n^2+\log(n)}{8} \\ 0 &\ge -4n^2+\log(n) \\ 0 &\ge -\log\left(2^{4n^2}\right)+\log(n) \\ 0 &\ge \log\left(\frac{n}{2^{4n^2}}\right) \\ 1 &\ge \frac{n}{2^{4n^2}} \end{aligned}$$

יודעים כי פונקציה אקספוננציאלית עם בסיס>1 גדלה הרבה יותר מהר מפולינום לכן מספיק להוכיח את אי השוויון עבור

$$\begin{aligned} 1 &\ge \frac{2^n}{2^{3n^2}} > \frac{n}{2^{4n^2}} \\ 1 &\ge \frac{2^n}{2^{3n^2}} \\ 1 &\ge \frac{1}{2^{3n^2-n}} \\ 2^{3n^2-n} &\ge 1 \\ 3n^2-n &\ge 0 \\ n(3n-1) &\ge 0 \\ n &\ge \frac{1}{3} \end{aligned}$$

לכן זה מתקיים עבור כל $n_0 \ge 2$. מש"ל.
אזי הוכחנו כי $n^2 = \Omega\left(n^2+\log(n)\right)$ וגם $n^2 = O\left(n^2+\log(n)\right)$ לכן $n^2 = \Theta\left(n^2+\log(n)\right)$

3 שאלה 3

צ"ל כי $n\log(n) = \Omega(n)$. כלומר צ"ל כי קיים $c>0$ שעבורו $\forall n\in\mathbb{N}, n>n_0$ מתקיים $n\log(n)\ge cn$.

$$\begin{aligned} n\log n &\geq cn \\ n\log n &\geq n \\ n\log n \underbrace{\geq}_{n\geq 4} 2n &\geq n \\ 2n &\geq n \\ n &\geq 0 \end{aligned}$$

$\log n$ פונקציה מונוטונית עולה ו $\log_2 4 = 2$ לכן עבור כל $n \geq 4$ מתקיים ש $\log_2 n \geq$ 2

$n \in \mathbb{N}$ לכן $n > 0$

4 שאלה 4

צ"ל כי $10\sqrt{n}+2 = O\left(\frac{n}{\log n}\right)$ כלומר צריך להוכיח כי קיים $c > 0$ כך ש לכל $n \in \mathbb{N}, n > n_0$ מתקיים $10\sqrt{n}+2 \leq \frac{cn}{\log n}$

יודעים כי לכל $n > 10$ מתקיים $n < \sqrt{n}$(ניתן להוכיח דרך גזירה ואינטגרציה) ו$\log n > 1$(כי זו פונקציה מונוטונית עולה ו $\log 10 > 1$) אזי אם נבחר $c = 20$ מקבלים:

$$\begin{aligned} 10\sqrt{n}+2 &\leq \frac{20n}{\log n} \\ 10\sqrt{n}+2 \leq 10\sqrt{n} &\leq \frac{20n}{\log n} \\ 10\sqrt{n} \leq 10n &\leq \frac{20n}{\log n} \\ 10n &\leq \frac{20n}{1} \leq \frac{20n}{\log n} \\ 10n &\leq 20n \\ 0 &\leq 10n \\ 0 &\leq n \end{aligned}$$

5 שאלה 5

הפתרון האכי קל הוא למצא a, b כך ש

$$\begin{aligned} n^{\log_b a} &= 2^{3\log_2 n} \\ n^{\log_b a} &= 2^{\log_2 n^3} \\ n^{\log_b a} &= n^3 \\ \log_b a &= 3 \end{aligned}$$

אזי $b = 2, a = 8$

ברור כי אם $f(x) = g(x)$ אזי גם $f(x) = \Theta(g(x))$(פשוט על ידי הצבת קבועים $\frac{1}{2}$ ו2)

6 שאלה 6

זה לא נכון כי לא מתקיים $\sqrt{\log\left(n^2\right)} = \Omega\left(\log n\right)$:

צריך להראות כי לא קיים $c > 0$ כך שעבור כל $n \in \mathbb{N}, n > n_0$ מתקיים $\sqrt{\log\left(n^2\right)} \geq c \log n$

$$\begin{aligned} \sqrt{\log\left(n^2\right)} &\geq c \log n \\ \sqrt{2} \cdot \sqrt{\log\left(n\right)} &\geq c \log n \end{aligned}$$

עבור $n > 2$ מתקיים $\log\left(n\right) > 0$ אזי

$$\begin{aligned} \sqrt{2} &\geq c\sqrt{\log n} \\ \sqrt{2} > 1 &\geq c\sqrt{\log n} \\ \frac{1}{\sqrt{\log n}} &\geq c \end{aligned}$$

אבל $\frac{1}{\sqrt{\log n}}$ לא חסום על ידי מספר $c > 0$ כי $\lim_{n\to\infty} \frac{1}{\sqrt{\log n}} = 0$. אזי הראינו כי לא קיים c כזה.

7 שאלה 7

It is impossible. When trying to get the last inserted item out of a stack all items below it have to be removed first(since we're using a queue), these items need to be saved somewhere, which takes O(n) memory.

8 שאלה 8

It is impossible. When trying to get the first inserted item out of a queue all items above it have to be removed first(since we're using a stack), these items need to be saved somewhere, which takes O(n) memory.

9 שאלה 9

9.1 צ״ל כי $f\left(n\right) = \Theta\left(h\left(n\right)\right)$

$f\left(n\right) = \Theta\left(g\left(n\right) + h\left(n\right)\right)$ וגם $g\left(n\right) = o\left(h\left(n\right)\right)$ כלומר:

$\lim_{n\to\infty} \frac{g(n)}{h(n)} \to 0$ אזי לכל $\epsilon > 0$ קיים n_0 כך שלכל $n > n_0$ שמקיים $\left|\frac{g(n)}{h(n)}\right| < \epsilon$

$$\begin{aligned} f\left(n\right) &= \Theta\left(\frac{\left(g\left(n\right) + h\left(n\right)\right) h\left(n\right)}{h\left(n\right)}\right) \\ &= \Theta\left(\frac{g\left(n\right)}{h\left(n\right)} h\left(n\right) + \frac{h^2\left(n\right)}{h\left(n\right)}\right) \\ &\underbrace{=}_{n>n_0} \Theta\left(\epsilon \cdot h\left(n\right) + h\left(n\right)\right) \\ & \Theta\left(\left(1+\epsilon\right) h\left(n\right)\right) \end{aligned}$$

9.2 צ"ל כי $f(n)=\Theta(g(n))$

נתון $f(n)=\Theta(g(n)+h(n))$ וגם $h(n)=O(g(n))$ כלומר $h(n)\le cg(n)$ אזי

$$\begin{aligned}
f(n) &= O(g(n)+h(n)) \\
&\le c_2(g(n)+h(n)) \\
&\le c_2(g(n)+cg(n)) \\
&\le c_2g(n)+c_2cg(n) \\
&\le (c_2+c_2c)\,g(n) \\
f(n) &= O(g(n))
\end{aligned}$$

$$\begin{aligned}
f(n) &= \Omega(g(n)+h(n)) \\
&\ge c_3(g(n)+h(n)) \\
&\ge c_3(g(n)+cg(n)) \\
&\ge c_3g(n)+c_3cg(n) \\
&\ge (c_3+c_3c)\,g(n) \\
&= \Omega(g(n))
\end{aligned}$$

9.3 זה לא נכון.

$$\begin{aligned}
n\log n &\le c\left(100n+n^{1+\frac{1}{\log n}}\right) \\
n\log n &\le 100cn+cn\cdot n^{\frac{1}{\log n}} \\
n\log n &\le cn\left(100+n^{\frac{1}{\log n}}\right) \\
n\log n &\le cn(100+n)<cn\left(100+n^{\frac{1}{\log n}}\right) \\
n\log n &\le cn^2+100cn
\end{aligned}$$

זה לא מתקיים. אזי הפרכנו.

10 שאלה 10

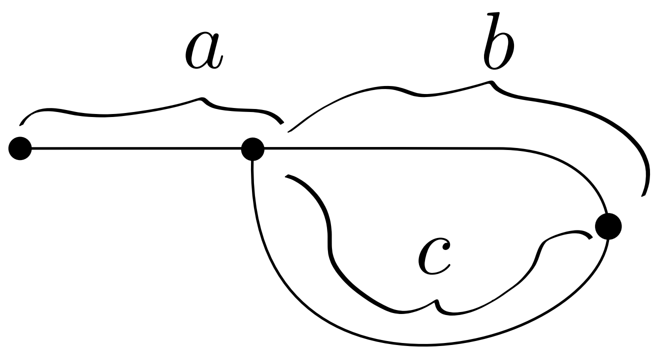
10.1 א

עוברים מ$head\to tail$ ובכל שלב בודקים אם כבר עברנו על ה$node$ הזה. פעולה זה עולה $O\left(n^2\right)$ (ברור, אל תשאלו למה כי אין לי זמן!)

אלגוריתם 1 תיקון הלולאה ב$O\left(n^2\right)$

```
FixLoop(head) {
–t=head;
–c=0;
–while(t!=null) {
—-c++
—-t=t.next();
—-t2=head;
—-c2=0;
——while(t2!=t) {
———c2++;
———prev=t2;
———t2=t2.next();
—-}
–}
–if(c!=c2) {//we found a loop
—-prev.next()=null;
—-return "fixed loop";
–}
–return "no loop found";
}
```

10.2 ב



איור 1: צורת הרשימה מקושרת

צריך למצוא node בתוך הלולאה, את זה עושים על ידי שתי מצביעים, אחד מהם מתקדם בצעדים של 1 והשני מתקדם בצעדים של 2. אם יש לולאה אזי שתי המצביעים יפגשו מתישהוא בתוך הלולאה(פעולה זו עולה $O\left(n\right)$)

אחרי שמצאנו נקודה בתוך הלולאה $n1$ מחשבים את אורך הלולאה ואת המרחק מ$head$ ל$n1$.

$$\begin{aligned} t1 = n1 \rightarrow head &= a + b \\ t2 = n_1 \rightarrow n \rightarrow n_1 &= c + b \end{aligned}$$

אזי $a-c=t_1-t_2$. אם נוכל להביא את ההפרש הזה ל0 אזי נקבל ששתי ה$nodes$ מרוחקים אותו מרחק מn. מביאים את זה לאפס על ידי קידום המצביע של n_1 או של $head$ בהתאם לסימן של t_1-t_2.
אזי אם הגענו לn אז יודעים כבר איך לתקן.

אלגוריתם 2 תיקון הלולאה בסיבוכיות $O(n)$

```
FixLoop(head)
–fast=slow=head;
–slow=slow.next();
–while(fast!=null && slow!=null && fast!=slow){
—-fast=fast.next().next();
—-slow=slow.next();
–}
–if(fast==null || slow==null) return "no loop found"
–n1=tmp=slow;
–h=tmp2=head;
–while(tmp2!=n1) {
—-t1++;
—-tmp2=tmp2.next()
–}
–t2=1;
–tmp=tmp.next();
—-while(tmp!=n1) {
—-t2++;
—-tmp=tmp.next()
–}
–while(t1-t2!=0) {
—-if(t1-t2>0) {
——h=h.next()
——t1–;
—-}else{
——prev=n1;
——n1=n1.next()
——t2–;
—-}
–}
–while(n1!=h){
—-prev=n1;
—-n1=n1.next();
—-h=h.next();
–}
–prev.next()=null
–return "loop fixed" //enjoy yourself
}
```