תרגיל מס.6

עפיף חלומה 302323001

7 במאי 2009

1 שאלה 1

ערימה היא עץ כמעט שלם, כלומר הרמה התחתונה ביותר היא הרמה היחידה הלא מלאה, והקודקודים בה תמיד מתחילים להתמלא משמאל לימין. כלומר אי אפשר שקודקוד i יהיה קיים בלי ש הקודקודים 0 עד $i-1$ בשורה הזו יהיו קיימים.

לפי זה, מקבלים מקסימום כאשר שני תת העצים יהיו שווים(כי לא יתכן שהעץ הימני יותר גדול מהשמאלי) אזי היחס הגדול ביותר הוא $\text{Left}/\text{Right} = 1$

2 שאלה 2

2.1 א

זה לא נכון עבור עץ בעל ערכים $1,1,1,1,1,1,1,1,1,1,1$.

עבל לא יתכן שכותב התרגילים יתן לנו שאלה כל כך טובה אזי אני אוכיח בהנחה שהערכים שונים.

לפי ההגדרה של ערימה:

לכל קודקוד i אם קיימים לו בנים a,b אזי $value\left(i\right) > value\left(a\right)$ ו $value\left(b\right)$.

איוור שהוא הSecond Largest כלומר יש רק איוור אחד שגדול ממנו. איוור זה צריך להיות ה Root כי הוא גדול מכולם.

לפי הגדרה זה יודעים כי כל קודקוד הוא גדול מכל בניו. אזי אם a,b הם הבנים של $Root$ מתקיים כי גם $\{a, Root\}$ גדולים מכל האיברים בתת העץ של a ו $\{b, Root\}$ גדולים מתת העץ השמאלי של $Root$. אזי יש רק שתי איברים שיש רק איבר אחד שיותר גדול מהם, אילו הם a,b. משל

2.2 ב

זה גם לא נכון על עץ בעל ערכים קבועים! מה אתם מנסים להטעה אותנו?

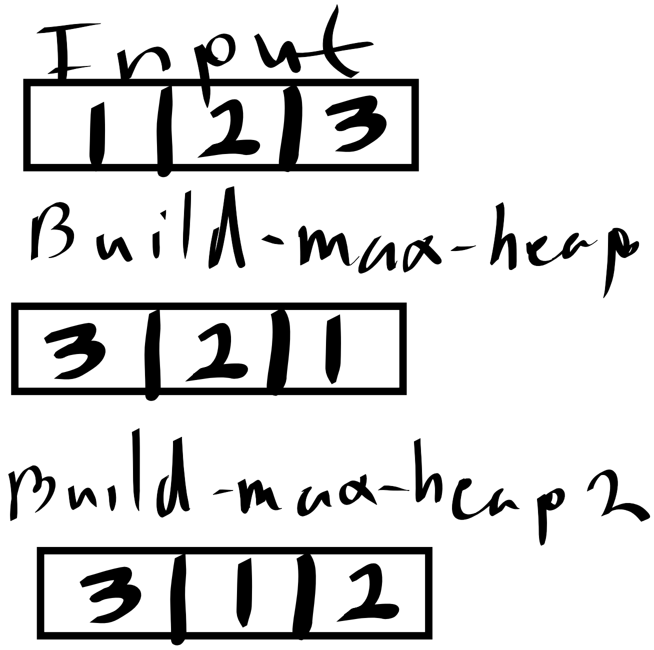
הוכחה עבור ערכים שונים

לפי ההגדרה אם עץ יש לו בנים אז הוא יותר גדול מהבנים שלו. אזי הבנים יותר קטנים ממנו. לפי כך כל קודקוד שיש לו בנים הוא לא הקטן ביותר. אזי הקודקוד הקטן ביותר אין לו בנים. אזי הקודקוד הקטן ביותר הוא עלה(לפי ההגדרה). משל.

3 שאלה 3

3.1 א

לא מקבלים אותה תוצאה. דוגמה:



איור 1: דוגמה לשאלה 3

3.2 ב

הפונקציה max-heap-insert עולה $\Theta\left(\log n\right)$ זמן תמיד. ואנחנו קוראים לה $n-1$ פעמים אזי זה $\Theta\left(\left(n-1\right)\log n\right) = \Theta\left(n \log n\right)$

4 שאלה 4

נראה לי שהוכחתי את זה בתרגיל הקודם....

יודעים כי עבור כל רמה(אם מתחילים לספור מ 0) יש 2^i קודקודים. אזי רוצים להוכיח כי

$$\sum_{i=0}^{n-1} 2^i = 2^n - 1$$

נשתמש באינדוקציה עבור n.

בדיקה: $n=0$ (הערה: הסכום לא מסכם כלום. מה הבעיה עם מתמטיקיים? הם לא מצינים החלק השלישי בלולאות הFor שלהם)

$$\begin{aligned}\sum_{i=0}^{-1}(\ldots) &= 2^0 - 1\\ 0 &= 1-1\\ 0 &= 0\end{aligned}$$

הנחת האינדוקציה: נניח כי ההנחה מתקיימת עבור $n = k$

צעד האינדוקציה: נוכיח עבור $n = k + 1$

$$\begin{aligned}
\sum_{i=0}^{k} 2^i &= 2^{k+1} - 1 \\
\sum_{i=0}^{k-1} 2^i + 2^k &= 2^{k+1} - 1 \\
\left(2^k - 1\right) + 2^k &= 2^{k+1} - 1 \\
2 \cdot 2^k - 1 &= 2^{k+1} - 1 \\
2^{k+1} - 1 &= 2^{k+1} - 1
\end{aligned}$$

משל.

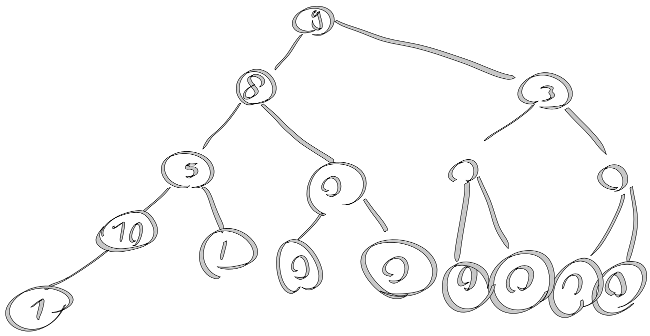
5 שאלה 5:

FindLVP	לחןעיפטם 1

```
FindLVP(a,b) {
current=0; //the head of the tree
while(true) {
–if(current>heap-size()) return "Nodes not in tree";
–if(a==value(current) OR b==value(current)) return current;
–cmp1=(a<value(current));
–cmp2=(b<value(current));
–if(cmp1!=cmp2) return current;
–else if(cmp1==true) current=Left(current);
–else current=Right(current);
}
}
```

6 שאלה 6

דוגמה נגדית:



איור 2: דוגמה נגדית

7 שאלה 7

7.1 א

זה לא מתקיים עבור עץ בינארי כלשהוא. העץ צריך להיות כמעט שלם[1]
נרצה להוכיח באינדוקציה כי עבור עץ בעל m עלים מתקיים

$$\frac{1}{m}\sum_{l\in leafs(T)} d\left(l\right)\geq\log\left(m\right)$$

בדיקה עבור $m=1$:

$$\begin{aligned} 1\cdot1 & \geq & \log\left(1\right)\\ 1 & \geq & 0 \end{aligned}$$

נניח כי זה מתקיים עבור כל $m=k$ אזי

$$\sum_{l\in leafs(T)} d\left(l\right)\geq k\cdot\log\left(k\right)$$

נוכיח עבור $m=k+1$ ונסמן העלה החדש ב q

$$\begin{aligned} \frac{1}{k+1}\sum_{l\in leafs(T\cup\{q\})} d\left(l\right) & \geq & \log\left(k+1\right)\\ \frac{\sum_{l\in leafs(T)} d\left(l\right)+d\left(q\right)}{k+1} & \geq & \log\left(k+1\right)\\ \frac{\sum_{l\in leafs(T)} d\left(l\right)+d\left(q\right)}{k+1}\geq\frac{k\cdot\log\left(k\right)+d\left(q\right)}{k+1} & \geq & \log\left(k+1\right)\\ k\cdot\log\left(k\right)+\underbrace{d\left(q\right)}_{\geq\log(k+1)} & \geq & \left(k+1\right)\cdot\log\left(k+1\right)\\ \left(k+1\right)\cdot\log\left(k\right) & \geq & \left(k+1\right)\cdot\log\left(k+1\right) \end{aligned}$$

[1] מעניין מאוד כמה הייתי מקבל על שאלה זו אם הייתי כותב כי הטענה לא נכונה ונותן דוגמה נגדית

זה לא יצא לי! אני יודע כי ההפרש בין $(\log(k+1) - \log(k))$ הולך וקטן וכנראה הוא נאבד איפה שהוא בקירובים שעשיתי... אבל אני לא רואה מה הייתי אמור לעשות.

7.2 ב

יודעים כבר כי אפשר לבנות אלגוריתם שיעבוד בסיבוכיות זמן $\Theta(n)$, גם אם זה לא תמיד יעיל מבחינת זכרון. ויודעים גם כי אי אפשר למיין מערך בלי לעבור על כל הערכים שלו אזי החסם המינימאלי זה $\Omega(n)$