מבני נתונים ומבוא לאלגוריתמים – פתרון מועד 99 מסמסטר ב2000

שאלה 1

א. נשתמש במשפט האב. בנוסחה זו: $a=2$, $b=2$, $n^{\lg_b a}=n^1=n$, $f(n)=(\lg n)^8$.

$f(n)=(\lg n)^8=O(n^{1-\varepsilon})$ (למשל, עבור $\varepsilon=0.5$), ולכן $T(n)=\Theta(n)$ (משפט האב, מקרה 1).

ב. נשתמש בשיטת האיטרציה:

$$T(n)=2T(n-1)+1=2[2T(n-2)+1]+1=4T(n-2)+3=4[2T(n-3)+1]+3=$$
$$=8T(n-3)+7=...=2^iT(n-i)+2^i-1=...=2^nT(0)+2^n-1=2^n-1$$

שימו לב שנוסחת הנסיגה מתארת את זמן הריצה של האלגוריתם הרקורסיבי לפתרון בעיית מגדלי האנוי.

שאלה 2

א. בשלב הראשון, השגרה מחשבת לכל איבר $A[i]$ את מספר האיברים **הגדולים** ממנו במערך (ועוד אחד), ומציבה מספר זה במשתנה x. לאחר מכן $A[i]$ מוצב במקום ה-x במערך B.

המערך B המתקבל הוא ממוין בסדר יורד, מפני שבמקום $B[1]$ נמצא האיבר הגדול ביותר ב-A, במקום $B[2]$ נמצא האיבר השני בגודלו ב-A וכך הלאה.

בשלב השני, השגרה מעתיקה את איברי B חזרה למערך A בסדר הפוך, ולכן מתקבל ב-A מערך ממוין.

ב. סיבוכיות הזמן של השגרה היא $O(n^2)$, מפני שהשלב הראשון בשגרה מורכב משתי לולאות מקוננות, שכל אחת מהן מתבצעת בדיוק n פעמים.

שאלה 3

א. **קלט ממוין בסדר עולה**: מיון-הכנסה יהיה היעיל ביותר, מפני שזמן הריצה שלו יהיה $O(n)$.

זמן הריצה של מיון-מהיר יהיה $\Theta(n^2)$ ושל מיון-ערימה $\Theta(n \lg n)$.

(זמן הריצה של מיון-ערימה הוא **תמיד** $\Theta(n \lg n)$ – ראו את תרגילים 6.4-4 ו- 6.4-5 בספר.)

ב. **קלט ממוין בסדר יורד**: מיון-ערימה יהיה היעיל ביותר – זמן ריצה של $\Theta(n \lg n)$.

זמני הריצה של מיון-הכנסה ושל מיון-מהיר יהיו $\Theta(n^2)$.

ג. **כאשר סופרים רק החלפות**:

קלט ממוין בסדר עולה: מיון-הכנסה יהיה היעיל ביותר – במהלך המיון לא תתבצע אף החלפה.

קלט ממוין בסדר יורד: מיון-ערימה יהיה היעיל ביותר – במהלך המיון יתבצעו $\Theta(n \lg n)$ החלפות. במיון-הכנסה יתבצעו $\Theta(n^2)$ החלפות.

במיון מהיר – בכל רמה זוגית של עץ הרקורסיה (החל מרמה 0) תתבצע החלפה אחת ויוחזר הערך $q=1$; בכל רמה אי-זוגית יתבצעו $r-p+1$ החלפות ויוחזר הערך $q=r$.

מספר ההחלפות הכולל יהיה לפיכך: $1+(n-1)+1+(n-3)+...+1=\Theta(n^2)$.

שאלה 4

א. אלגוריתם איטרטיבי הבודק אם עץ בינרי T הוא עץ חיפוש בינרי:

CHECK-BST (T)

$x \leftarrow$ TREE-MINIMUM (T)

$i \leftarrow 1$, $bst \leftarrow$ TRUE

while $i \leq n - 1$ and bst = TRUE

    do $y \leftarrow$ TREE-SUCCESSOR (x)

        if key[x] $\leq$ key[y]

            then $i \leftarrow i + 1$

                $x \leftarrow y$

            else $bst \leftarrow$ FALSE

return bst

לפי תרגיל 12.2-7 בספר, זמן הריצה של האלגוריתם הוא $\Theta(n)$.

הערה: אפשר גם לבצע סריקה תוכית רגילה של העץ ולבדוק אם המערך המתקבל הוא ממוין.

ב. אלגוריתם רקורסיבי הבודק אם עץ בינרי T מקיים את תכונת הערימה:

CHECK-HEAP (T)

if T = NIL

    then return TRUE

    else if (left[T] $\neq$ NIL and key[left[T]] > key[T]) or

            (right[T] $\neq$ NIL and key[right[T]] > key[T])

            then return FALSE

            else return CHECK-HEAP (left[T]) and CHECK-HEAP (right[T])

בכל צומת בעץ מתבצעות שתי קריאות רקורסיביות ולכן זמן הריצה הוא $\Theta(n)$.

שאלה 5

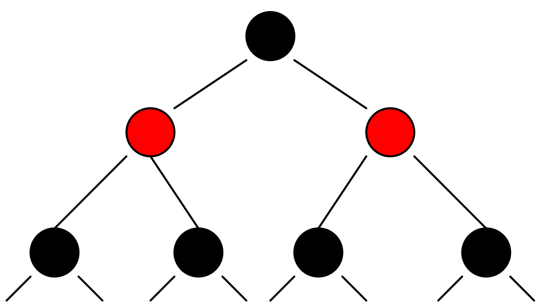
א. הגובה של עץ אדום-שחור המכיל שבעה צמתים פנימיים יהיה מינימלי כאשר העץ יהיה עץ בינרי **שלם**. גובה העץ במקרה זה הוא $3 = \lfloor \lg_2 15 \rfloor = \lfloor \lg_2 n \rfloor$. (ראו את הציור בסעיף ב').

n מציין כאן את מספר כל צמתי העץ, כולל העלים.

הגובה המקסימלי של עץ אדום-שחור המכיל שבעה צמתים פנימיים הוא 4 (ראו בסעיף ב').

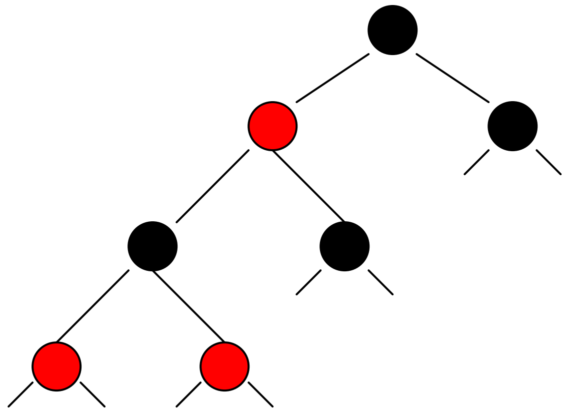
גובה העץ לא יכול להיות גדול מ-4, מפני שאז התנאי הנזכר בתרגיל 13.1-5 בספר לא יתקיים עבור השורש.

ב. דוגמה לעץ אדום-שחור המכיל שבעה צמתים פנימיים שגובהו מינימלי:



גובה העץ: 3

דוגמה לעץ אדום-שחור המכיל שבעה צמתים פנימיים שגובהו מקסימלי :



גובה העץ : 4