5.6.2015

מבני נתונים
פתרון תרגיל מס׳ 6

1. שאלה זו הופיעה במבחן מועד א 2014

כתוב/כתבי פסאודו-קוד של פונקציה בשם P1, יעילה ככל האפשר, אשר מקבלת כפרמטר עץ בינארי T ומדפיסה עבור כל צומת x בעץ את המפתח של הצומת ולאחריו מספר שמציין את מספר הצמתים שנמצאים במסלול שמחבר בין צומת x לצומת בעל המפתח הגדול ביותר בתת העץ של T ששורשו x (תת העץ כולל גם את x).

אין חשיבות לסדר ההדפסה של הצמתים בפלט.

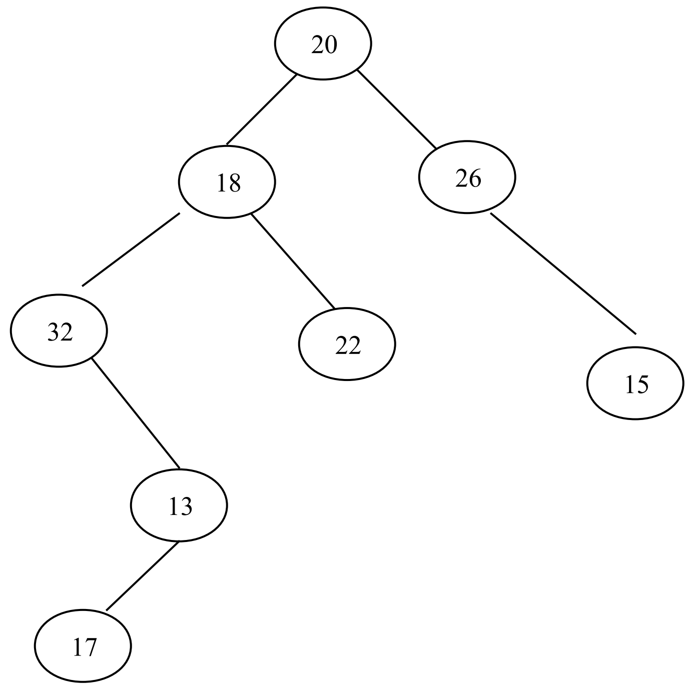
נתח/י את סיבוכיות זמן הריצה של הפונקציה שכתבת כתלות במספר הצמתים בעץ n.

הנחות ודרישות:

- **אין להשתמש במבני עזר נוספים.**
- **מותר להשתמש במספר קבוע של משתנים (כמו למשל x,y,z).**
- **מותר להשתמש בפונקציות עזר, אך יש לכתוב את הפסיאודו-קוד של פונקציות העזר.**
- **כל צומת x בעץ מכיל את השדות הרגילים של עץ בינארי כפי שהוגדר בכיתה.**

<u>דוגמה:</u>

יהי T עץ בינארי שמתואר בציור הבא :



לאחר הקריאה לפונקציה P1(T) יתקבל הפלט:

```
20 3 18 2 26 1 32 1 22 1 15 1 13 2 17 1
```

הסבר לפלט: עבור צומת 20 הודפס המספר 3 כי המסלול בין 20 ל- 32 מכיל 3 צמתים.
עבור צומת 26 הודפס המספר 1 כי המסלול בין 26 לעצמו מכיל צומת אחת. וכן הלאה.

פתרון

הערה: לא ראיתי שמישהו פתר נכון את השאלה הזו. הבעיה בפתרון היתה שבמקרה של עץ ריק זה לא נכון להחזיר ערך מקסימאלי 0. (כי לא כתוב בשאלה שמותר להניח שכל המפתחות הם מספרים חיוביים או 0).

P1(T)

```
x=root(T)
if (x==NULL) {return}
y.p=1
y.max=key(x)
if (left(x)==NULL) && (right(x)==NULL) { print key(x),y.p
                                          return y}
if (left(x)!=NULL) { y1=P1(T_left(x)) }
if (right(x)!=NULL) { y2=P1(T_right(x)) }
if (left(x)==NULL) { if (key(x) > y2.max) { print key(x),y.p
                                             return y}
                y.p=y2.p+1
                y.max=y2.max
                print key(x),y.p
                return y
               }
if (right(x)==NULL) { if (key(x) > y1.max){ print key(x),y.p
                                              return y}
                y.p=y1.p+1
                y.max=y1.max
                print key(x),y.p
                return y
               }
y.max=max{y.max, y1.max, y2.max}
if (y.max==key(x)) {print key(x),y.p ; return y}
if (y.max==y1.max) {y.p=y1.p+1}
else {y.p=y2.p+1}
print key(x),y.p
return y
```

לניתוח הסיבוכיות, לכל צומת **x** (פרט לשורש) יש בדיוק קריאה רקורסיבית אחת כשהפרמטר לתוכנית הוא העץ ששורשו **x** (דהינו T_x). לכן מספר הקריאות הרקורסיביות הוא בדיוק **1-n**. מכאן המשך ניתוח הסיבוכיות הוא כמו בשאלה 2 בהחלפת מספר הקריאות הרקורסיביות ב- **1-n** (במקום **2n**).

2. שאלה זו הופיעה במבחן מועד ב 2014

הגדרה: נגדיר עץ בינארי עם צבעים אדום-לבן, כעץ בינארי שבו לכל צומת x יש בנוסף לשדות הרגילים, שדה color(x) ששווה ל- red אם הצומת אדום או white אם הצומת לבן.

הגדרה: עבור צומת x בעץ בינארי T נגדיר את תת העץ השמאלי של x כתת העץ של T שמורכב מהבן השמאלי של x וכל הצמתים שמתחתיו. במילים אחרות, תת העץ השמאלי של x שווה ל- $T_{left(x)}$ וכולל גם את left(x). באופן דומה נגדיר את תת העץ הימני של x כתת העץ של T שמורכב מהבן הימני של x וכל הצמתים שמתחתיו.

הגדרה: נגדיר שצומת x בעץ בינארי עם צבעים אדום-לבן T הוא צומת טוב אם סכום הגבהים של כל הצמתים הלבנים בתת העץ השמאלי של x גדול יותר מסכום הגבהים של כל הצמתים האדומים בתת העץ הימני של x.

כתוב/כתבי פסאודו-קוד של פונקציה בשם P1, יעילה ככל האפשר, אשר מקבלת כפרמטר עץ בינארי עם צבעים אדום-לבן T ומדפיסה את המפתחות של כל הצמתים הטובים בעץ T.

אין חשיבות לסדר ההדפסה של הצמתים בפלט.

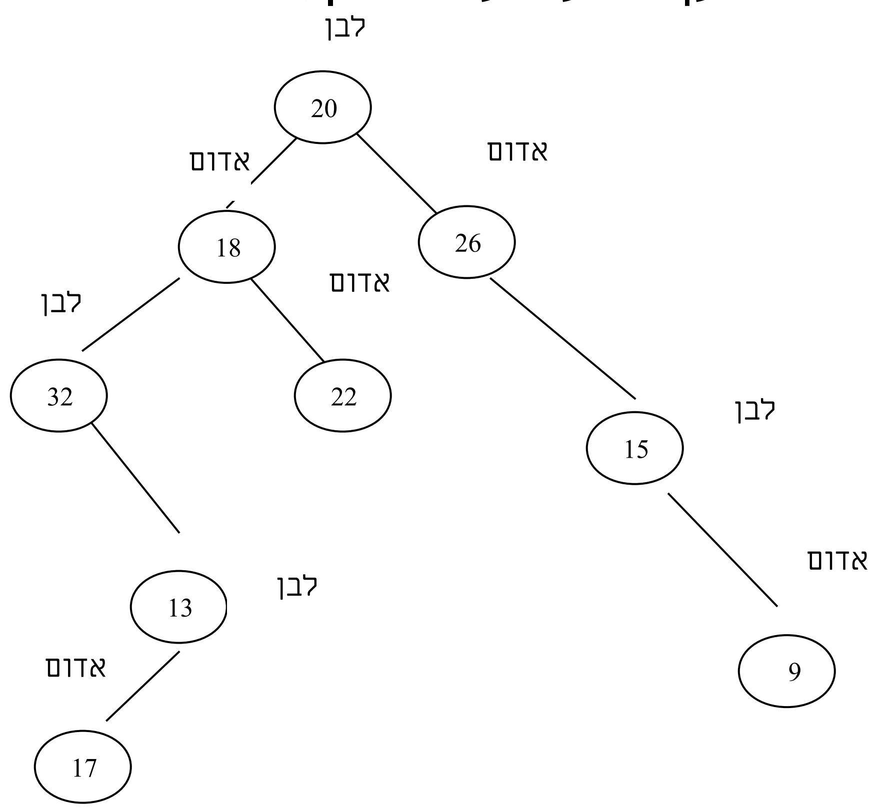
נתח/י את סיבוכיות זמן הריצה של הפונקציה שכתבת כתלות במספר הצמתים בעץ n.

הנחות ודרישות:

- **אין להשתמש במבני עזר נוספים.**
- **מותר להשתמש במספר קבוע של משתנים (כמו למשל x,y,z).**
- **מותר להשתמש בפונקציות עזר, אך יש לכתוב את הפסיאודו-קוד של פונקציות העזר.**
- **כל צומת x בעץ מכיל את השדות הרגילים של עץ בינארי כפי שהוגדר בכיתה ובנוסף מכיל את השדה color(x) שתואר לעיל.**

<u>דוגמה:</u>

יהי T עץ בינארי עם צבעים אדום-לבן שמתואר בציור הבא :



לאחר הקריאה לפונקציה P1(T) יתקבל הפלט:

```
20 18
```

הסבר לפלט: עבור צומת 20 סכום הגבהים של הצמתים הלבנים בתת העץ השמאלי שווה ל- 5
וסכום הגבהים של הצמתים האדומים בתת העץ הימני שווה ל- 4 ולכן צומת 20 הוא טוב.

עבור צומת 18 סכם הגבהים של הצמתים הלבנים בתת העץ השמאלי שווה ל- 5 וסכום הגבהים של הצמתים האדומים בתת העץ הימני שווה ל- 1 ולכן צומת 18 הוא צומת טוב.

עבור צומת 32 סכום הגבהים של הצמתים הלבנים בתת העץ השמאלי שווה ל-0 וסכום הגבהים של הצמתים בתת העץ הימני שווה ל-1 ולכן צומת 32 אינו צומת טוב.

<u>פתרון</u>

```
P2(T)
x=root(T)
if (x==NULL) { y.h=0
               y.wsum=0
               y.rsum=0
               return y
              }
y1=P2(T_left(x))
y2=P2(T_right(x))
y.h=max {y1.h, y2.h} + 1
y.wsum=y1.wsum + y2.wsum
if (color(x)==white) { y.wsum += y.h }
if (color(x)==red) { y.rsum += y.h }
if (y1.wsum > y2.rsum} { print key(x) }
return y
```

ננתח את הסיבוכיות באופן הבא:

נסמן ב-x את מס' הקריאות הרקורסיביות. מאחר ועבור כל צומת בעץ מתבצעות שתי קריאות רקורסיביות (אחת לבן השמאלי שלו ואחת לבן הימני שלו). נקבל ש $x=2n$.

נסמן ב- C_1 את מס' הפעולות שמופיעות בתכנית ללא אלה שמתבצעות בקריאות הרקורסיביות.

מצד אחד נקבל:

$$T(n) \le C1 + C1 \cdot x = C1 + C1 \cdot 2n = \theta(n)$$

ולכן נקבל: $T(n) = O(n)$

מצד שני מאחר וכל קריאה רקורסיבית מבצעת לפחות פעולה אחת נקבל:

$$T(n) \ge x = 2n = \theta(n)$$

ולכן נקבל: $T(n) = \Omega(n)$

3. שאלה זו הופיעה במבחן מועד ג 2014

<u>הגדרה</u>: נגדיר <u>עץ בינארי עם צבעים אדום-לבן</u>, כעץ בינארי שבו לכל צומת x יש בנוסף לשדות הרגילים, שדה color(x) ששווה ל- red אם הצומת אדום או white אם הצומת לבן.

כתוב/כתבי פסאודו-קוד של פונקציה בשם P1, <u>יעילה ככל האפשר</u>, אשר מקבלת כפרמטר עץ בינארי עם צבעים אדום-לבן T ומדפיסה עבור כל צומת x בעץ את key(x) ולאחריו את מספר הצמתים במסלול הארוך ביותר שמתחיל מהצומת x לצומת לבן שנמצא בתת העץ ששורשו x.

אין חשיבות לסדר ההדפסה של הצמתים בפלט.

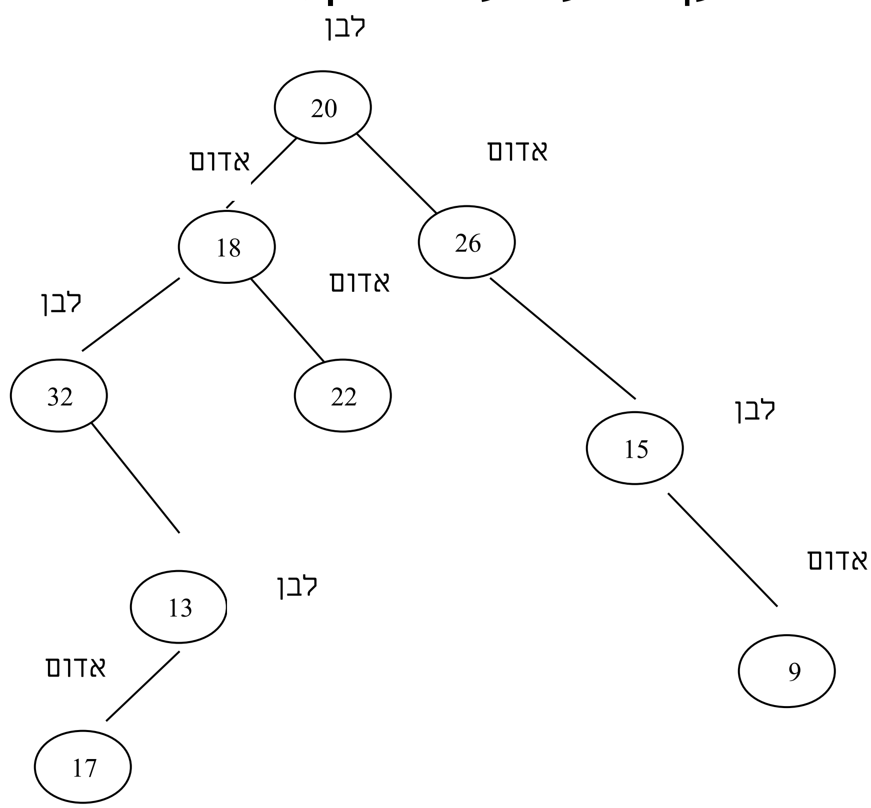
נתח/י את סיבוכיות זמן הריצה של הפונקציה שכתבת כתלות במספר הצמתים בעץ n.

<u>הנחות ודרישות:</u>

- **אין להשתמש במבני עזר נוספים.**
- **מותר להשתמש במספר קבוע של משתנים (כמו למשל x,y,z).**
- **מותר להשתמש בפונקציות עזר, אך יש לכתוב את הפסיאודו-קוד של פונקציות העזר.**
- **כל צומת x בעץ מכיל את השדות הרגילים של עץ בינארי כפי שהוגדר בכיתה ובנוסף מכיל את השדה color(x) שתואר לעיל.**

<u>דוגמה:</u>

יהי T עץ בינארי עם צבעים אדום-לבן שמתואר בציור הבא :



לאחר הקריאה לפונקציה P1(T) יתקבל הפלט:

```
20 4   18 3   32 2   13 1   17 0   22 0   26 2   15 1   9   0
```

הסבר לפלט: עבור צומת 20 המסלול הארוך ביותר, שמתחיל מצומת 20 לצומת לבן בתת העץ ששורשו 20 הוא המסלול אל הצומת 13 והוא מכיל 4 צמתים ולכן הודפס הזוג: 4 20.

עבור צומת 26 המסלול הארוך ביותר, שמתחיל בצומת 26 לצומת לבן בתת העץ ששורשו 26 הוא המסלול אל צומת 15 והוא מכיל 2 צמתים ולכן הודפס הזוג: 2 26.

עבור צומת 17 מאחר ואין צמתים לבנים בתת העץ ששורשו 17, המסלול הארוך ביותר, שמתחיל בצומת 17 לצומת לבן בתת העץ ששורשו 17 לא קיים ולכן הוא מכיל 0 צמתים, ולכן הודפס הזוג 0 17.

פתרון

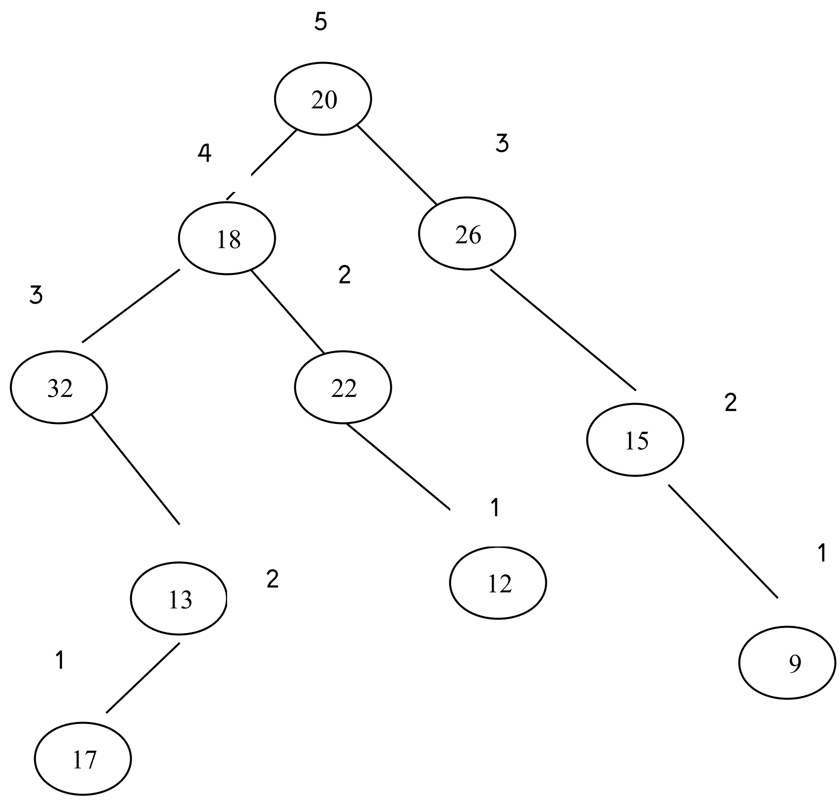
```
P3(T)
x=root(T)
if (x==NULL) { y=0
               return y
              }
y1=P3(T_left(x))
y2=P3(T_right(x))
if (color(x)==red && y1==0 && y2==0) { y=0 }
else { y=max{y1,y2}+1 }
return y
```

ניתוח הסיבוכיות זהה לניתוח הסיבוכיות של שאלה 2.

<u>4. שאלה זו הופיעה במבחן מועד ב 2006</u>

נגדיר "עץ בינארי עם גבהים" כעץ בינארי T שבו לכל צומת x בנוסף לשדות הרגילים יש שדה h(x) המכיל את גובה תת העץ ששורשו x.

לדוגמה: יהי T העץ הבא שבו לכל צומת מצוין מפתח הצומת ומעליו מספר המציין את גובה הצומת.



לכל איבר x בעץ בינארי עם גבהים ישנם השדות הבאים:
info(x),key(x),left(x),right(x),parent(x),h(x)

בנוסף ישנו שדה root(T) שמצביע לשורש העץ T.

נגדיר שצומת y הוא <u>שכן ימני</u> של צומת x אם x ו- y נמצאים באותה רמה בעץ (דהינו באותו מרחק מהשורש) ו- y הוא הצומת הראשון שנמצא מימין ל- x ברמה של x. לדוגמה, 15 שכן ימני של 22 אבל 15 אינו שכן ימני של 32.

כתוב/י פסאודו-קוד של פונקציה <u>יעילה ביותר</u> בשם P2(T ,k) שמקבלת כפרמטרים עץ בינארי עם גבהים T ו- מספר k. הפונקציה מדפיסה yes אם קיימים שני צמתים x ו- y בעץ כך ש- y הוא שכן

ימני של **x** ואורך המסלול שמחבר בין **x** ל- **y** בעץ הוא **k**. אחרת, הפונקציה תדפיס no.

<u>דוגמאות:</u>

לאחר הקריאה לפונקציה P2(T,3) כאשר T הוא העץ שבציור הנ"ל הפונקציה תדפיס yes. למשל בגלל הצמתים 22 ו- 18 והמסלול 32-18-22 המחבר אותם.

לאחר הקריאה לפונקציה P2(T,4) כאשר T הוא העץ שבציור הנ"ל הפונקציה תדפיס no כי לא קיימים שני צמתים שהאחד שכן ימני של השני והמרחק בינם הוא 4.

לאחר הקריאה לפונקציה P2(T,5) כאשר T הוא העץ שבציור הנ"ל הפונקציה תדפיס yes בגלל הצמתים 13 ו- 21 והמסלול

13-32-18-22-21 המחבר אותם.

<u>פתרון</u>

לצורך הפתרון נגדיר ש- h(NULL)=0 ואז כשאנחנו כותבים בתכנית למשל h(left(x)) אנחנו לא צריכים לבדוק לפני כן ש- left(x) אינו NULL.

```
P4(T,k)
y=P5(T,k)
if (y==true) {print "yes"}
else {print "no"}

P5(T,k)
x=root(T)
if (x==NULL || k<3 || k%2==0) { y=false
                                 return y
                               }
y1=P5(T_left(x),k)
y2=P5(T_right(x),k)
if (y1==true || y2==true) { y=true
                            return y
                          }
if (1+2*min{h(left(x),h(right(x)}==k) {y=true}
else {y=false}
return y
```

לניתוח הסיבוכיות, בגלל שבמקרה הגרוע ביותר מכל צומת יהיו שתי קריאות רקורסיביות, הניתוח זהה לניתוח של שאלה 2.