

24.5.2016

מבני נתונים  
פתרון תרגיל מס' 7

**1. שאלה זו הופיע במבחן מועד א בסמסטר ב תשע"ה**

נזכיר שעבור עץ  $T$  וצומת  $x$  בעץ, תת העץ של  $T$  ששורשו  $x$  מסומן ב- $T_x$  (תת העץ הזה כולל גם את  $x$ ).

**הגדרה:** נגדיר שצומת  $x$  בעץ בינארי  $T$  הוא צומת טוב, אם סכום המפתחות של כל העלים בתת העץ ששורשו  $x$  (דהינו בתת העץ  $T_x$ ) גדול מסכום המפתחות של כל הצמתים שנמצאים במסלול שמחבר את  $x$  לשורש העץ  $T$  (המסלול כולל גם את  $x$  וגם את שורש העץ  $T$ ).

לדוגמה, בעץ שבצירור שבעמוד הבא, הצומת 26 הוא צומת טוב כי סכום העלים בתת העץ  $T_{26}$  שווה ל-  $30+19=49$  והוא גדול מסכום הצמתים במסלול שמחבר את 26 לשורש העץ ששוה ל-  $26+20=46$ .

כתוב/כתבי פסאודו-קוד של פונקציה בשם  $P1$ , יעילה ככל האפשר, אשר מקבלת כפרמטר עץ בינארי  $T$  ומדפיסה עבור כל צומת טוב בעץ  $x$  את המפתח של הצומת  $x$  ולאחריו מספר ששוה למספר הצמתים שנמצאים במסלול הארוך ביותר שמחבר בין הצומת  $x$  לאיזשהו צומת טוב שנמצא בתת העץ ששורשו  $x$  (דהינו בתת העץ  $T_x$ ).

אין חשיבות לסדר ההדפסה של הצמתים בפלט.

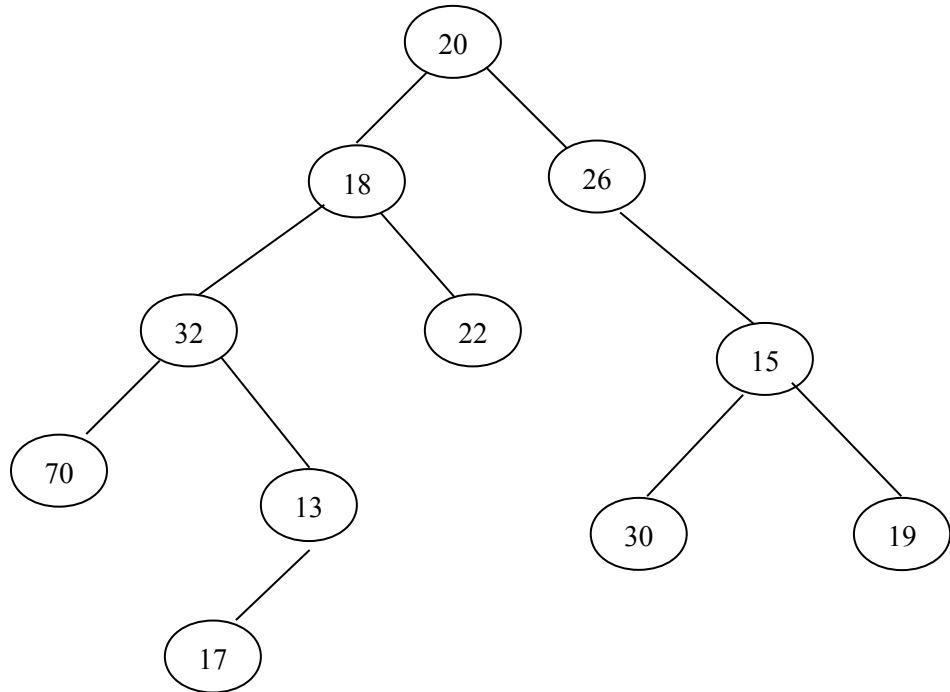
נתח/י את סיבוכיות זמן הריצה של הפונקציה שכתבת כתלות במספר הצמתים בעץ  $n$ .

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 הפסאודו-קוד של פונקציות העזר.
- כל צומת  $x$  בעץ מכיל את השדות הרגילים של עץ בינארי כפי שהוגדר בכיתה.

דוגמה:

יהי  $T$  עץ בינארי שמתואר בציור הבא :



לאחר הקריאה לפונקציה  $P1(T)$  יתקבל הפלט:

26 1 20 3 18 2 32 1

הסבר לפלט: הצמתים הטובים בעץ הם צמתים 26,20,18,32. לדוגמה, צומת 26 הוא צומת טוב כי סכום העלים בתת העץ  $T_{26}$  שווה ל-  $30+19=49$  והוא גדול מסכום הצמתים במסלול שמחבר את 26 לשורש העץ ששוה ל-  $26+20=46$ . מאחר ובתת העץ  $T_{26}$  אין צומת טוב נוסף, מספר הצמתים במסלול הארוך ביותר המחבר בין 26 לצומת טוב בתת העץ  $T_{26}$  שווה ל-1.

באופן דומה, צומת 18 הוא צומת טוב, ובתת העץ  $T_{18}$  הצמתים הטובים הם 18,32. לכן המסלול הארוך ביותר המחבר בין צומת 18 לצומת טוב בתת העץ  $T_{18}$  מכיל את הצמתים 18 ו-32 ומספר הצמתים במסלול הזה הוא 2.

P1(T)

P1.1(T,0)

P1.1(T,s)

x=root(T)

if (x==null) {y.distance\_to\_good=0

    y.sum\_leaves=0

    return y }

if (left(x)==null && right(x)==null) {

    y.distance\_to\_good=0

    y.sum\_leaves=key(x)

    return y}

y1=P1.1(T<sub>left</sub>(x),s+key(x))

y2=P1.1(T<sub>right</sub>(x),s+key(x))

y.sum\_leaves=y1.sum\_leaves+y2.sum\_leaves

good=false

if (y.sum\_leaves > s+key(x)) {good=true}

n=max{y1.distance\_to\_good,y2.distance\_to\_good}

if (n>0) {n++}

if (n==0 && good==true) {n=1}

y.distance\_to\_good=n

return y

## 2. שאלה זו הופיע במבחן מועד ב בסמסטר ב תשע"ה

הגדרה: נגדיר עץ בינארי אדום לבן, כעץ בינארי שבו לכל צומת  $x$ , בנוסף לשדות הרגילים, יש שדה צבע  $color(x)$  שהוא אדום (red) או לבן (white).

הגדרה: נגדיר שמסלול  $P$  בעץ בינארי אדום לבן הוא מסלול מתחלף אם אין במסלול שני צמתים עוקבים עם צבע זהה.

לדוגמה, המסלול שמחבר בין הצמתים 19 ו-20 בעץ שבצירור בעמוד הבא הוא מסלול מתחלף, והמסלול שמחבר בין הצמתים 26 ו-22 אינו מסלול מתחלף.

כתוב/כתבי פסאודו-קוד של פונקציה בשם  $P2$ , יעילה ככל האפשר, אשר מקבלת כפרמטר עץ בינארי אדום לבן  $T$  ומדפיסה את מספר הצמתים במסלול המתחלף הארוך ביותר שמחבר בין שני צמתים בעץ.

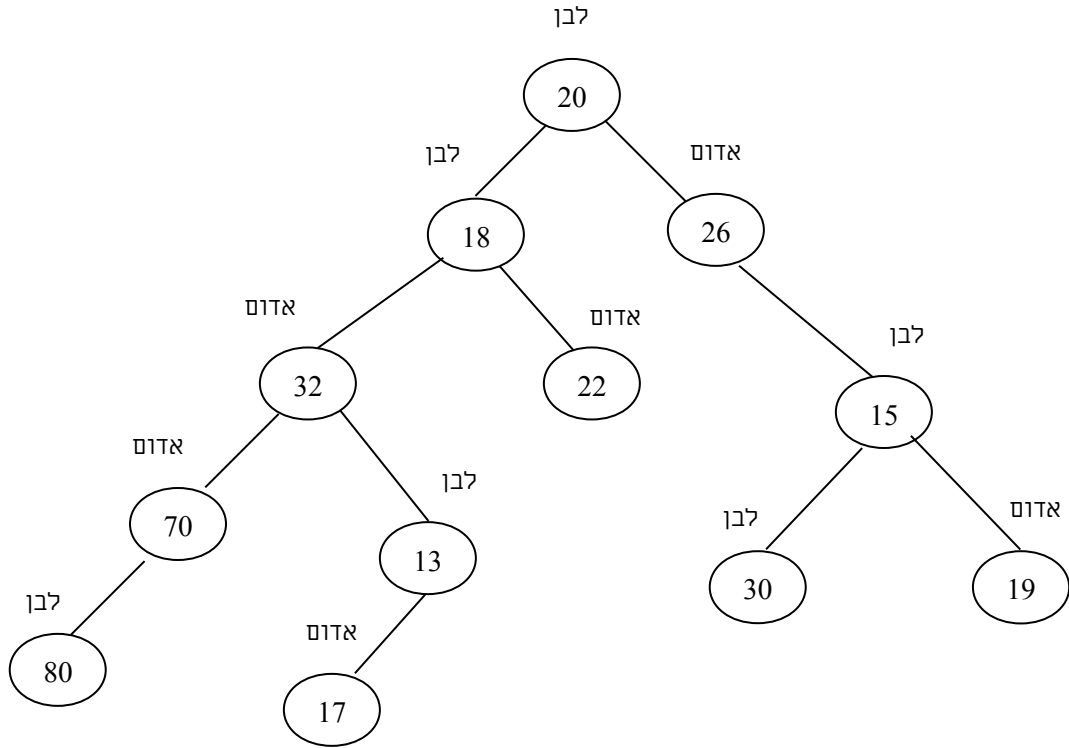
נתח/י את סיבוכיות זמן הריצה של הפונקציה שכתבת כתלות במספר הצמתים בעץ  $n$ .

### 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 את השדה  $color(x)$  שתואר למעלה.

דוגמה:

יהי  $T$  עץ בינארי אדום לבן שמתואר בציור הבא :



לאחר הקריאה לפונקציה  $P2(T)$  יתקבל הפלט: 5

הסבר לפלט: המסלול המתחלף הארוך ביותר בעץ הוא המסלול הבא  
17 13 32 18 22 :22 - ל- 17

מסלול זה מכיל 5 צמתים ולכן הפלט הוא 5.

P2(T)

y=P2.1(T)

print y.max\_dist

P2.1(T)

x=root(T)

if (x==null) {y.dist\_from\_root=0

    y.max\_dist=0

    return y}

y1=P2.1(T<sub>left</sub>(x))

y2=P2.1(T<sub>right</sub>(x))

n1=1

n2=1

if (left(x)!=null && color(x)!=color(left(x))) {

    n1=y1.dist\_from\_root+1}

}

if (right(x)!=null && color(x)!=color(right(x))) {

    n2=y2.dist\_from\_root+1}

}

y.dist\_from\_root=max{n1,n2}

n=n1+n2-1

y.max\_dist=max{y1.max\_dist,y2.max\_dist,n}

return y

### 3. שאלה זו הופיע במבחן מועד ג בסמסטר ב תשע"ה

הגדרה: נגדיר ש-  $T$  הוא עץ בינארי אריתמטי אם  $T$  הוא עץ בינארי שבו לכל צומת  $x$  בעץ בנוסף לשדות הרגילים יש שני שדות נוספים:

$val(x)$  - מכיל את הערך של  $x$  שיכול להיות מספר שלם כלשהו או פעולה אריתמטית כלשהי.

$type(x)$  - מכיל את הסוג של  $x$  שיכול להיות  $n$  (אם  $x$  הוא מספר) או  $o$  (אם  $x$  הוא פעולה אריתמטית).

הגדרה: נגדיר ש-  $T$  הוא עץ בינארי אריתמטי חוקי אם  $T$  הוא עץ בינארי אריתמטי שמקיים את התנאים הבאים:

(1) כל העלים בעץ הם מספרים (דהינו צמתים מסוג  $n$ ) וכל הצמתים בעץ שאינם עלים הם פעולות אריתמטיות (דהינו צמתים מסוג  $o$ ).

(2) לכל צומת שאינו עלה בעץ יש בדיוק שני בנים (דהינו אסור שיהיה לו רק בן אחד).

כתוב/כתבי פסאודו-קוד של פונקציה בשם  $P3$ , יעילה ככל האפשר, אשר מקבלת עץ בינארי אריתמטי  $T$  ומדפיסה  $yes$  אם העץ הוא עץ בינארי אריתמטי חוקי או  $no$  אם העץ אינו עץ בינארי אריתמטי חוקי.

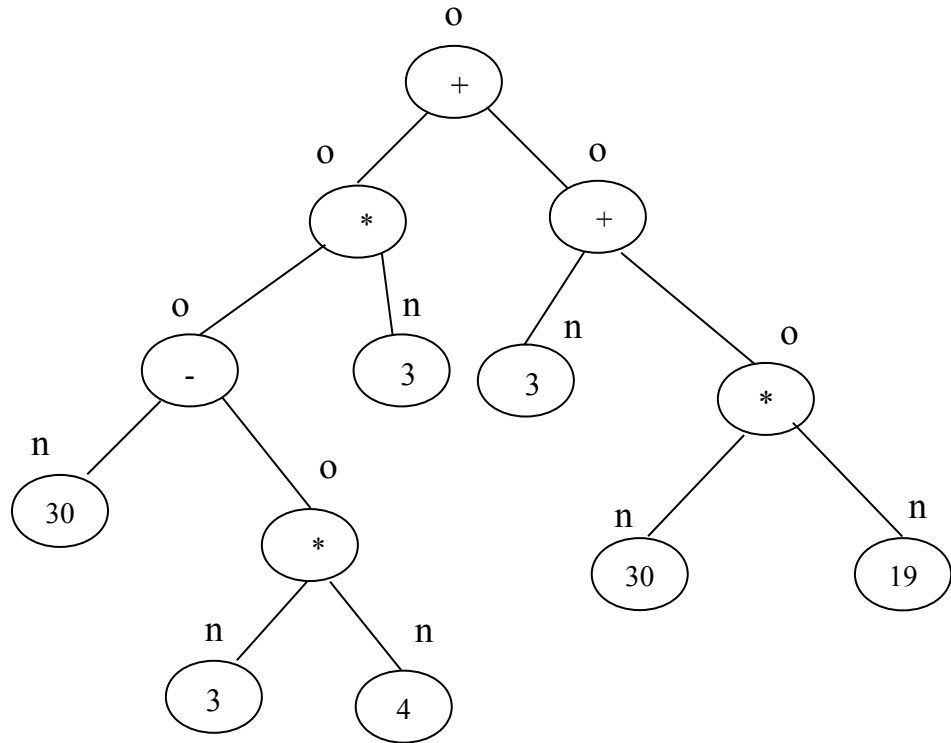
נתח/י את סיבוכיות זמן הריצה של הפונקציה שכתבת כתלות במספר הצמתים בעץ  $n$ .

#### 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 את השדות  $value(x)$  ו-  $type(x)$  שתוארו למעלה.

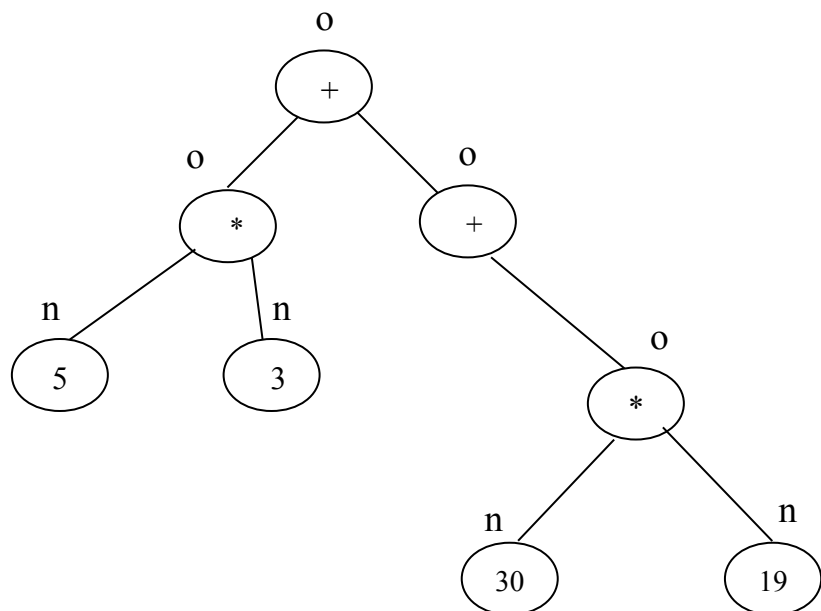
דוגמה:

יהי  $T_1$  עץ בינארי אריתמטי שמתואר בציור הבא, כאשר עבור כל צומת  $x$  בעץ מתואר הערך של  $val(x)$  ומעליו הערך של  $type(x)$ :



לאחר הקריאה לפונקציה  $P3(T_1)$  יתקבל הפלט: **yes**

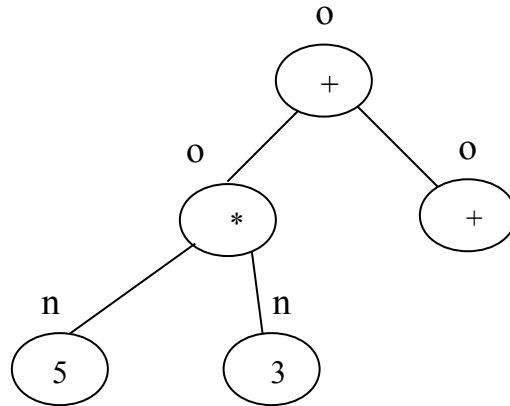
יהי  $T_2$  עץ בינארי אריתמטי שמתואר בציור הבא:





לאחר הקריאה לפונקציה  $P3(T_1)$  יתקבל הפלט: no

יהי  $T_3$  עץ בינארי אריתמטי שמתואר בציור הבא:



לאחר הקריאה לפונקציה  $P3(T_3)$  יתקבל הפלט: no

P3(T)

```
y=P3.1(T)
if (y==true) print {yes}
else {print no}
```

P3.1(T)

```
x=root(T)
if (x==null) {return true}
y1=P3.1(Tleft(x))
y2=P3.1(Tright(x))
if (y1==false || y2==false) {return false}
if (type(x)==o) {
  if (left(x)=null || right(x)==null) {return false}
  return true
}
if (type(x)=n) {
  if (left(x)!=null || right(x)!=null) {return false}
  return true
}
```