



מכללת אורט כפר-סבא

## מבני נתונים ויעילות אלגוריתמים

### תרגיל מס' 12

פתרו את השאלות הבאות. יש לסיים את התרגיל עד יום ד' (31.12).

#### שאלה 1

ממשו את הפונקציה האיטרטיבית (לא רקורסיבית) הבאה:

```
int find_product (int a[], int n, int x)
```

הפונקציה מקבלת מערך  $a$  המכיל  $n$  מספרים שלמים חיוביים (טבעיים) והממוין בסדר עולה, וכן מספר טבעי  $x$ . הפונקציה תחזיר 1 אם ניתן להציג את  $x$  כמכפלה של שני איברים מהמערך  $a$ , ו-0 אם לא ניתן להציג את  $x$  כמכפלת שני איברים מהמערך  $a$ .

לדוגמא: עבור המערך  $a = \{2,2,3,5,8,9\}$  שגודלו  $n = 6$ , הקריאה  $\text{find\_product}(a,6,18)$  תחזיר 1, שכן קיימים זוג איברים במערך שמכפלתם שווה ל-18 (מדובר באיברים 2 ו-9), והקריאה  $\text{find\_product}(a,6,24)$  תחזיר 1 שכן מכפלת האיברים 3 ו-8 שווה ל-24.

לעומת זאת, עבור הקריאה  $\text{find\_product}(a,6,25)$  יוחזר הערך 0 שכן אין שני איברים במערך שמכפלתם שווה ל-25. גם עבור הקריאה  $\text{find\_product}(a,6,11)$  יוחזר הערך 0.

כתבו את הפונקציה  $\text{find\_product}$  כך שסיבוכיות זמן הריצה שלה תהיה  $\Theta(n)$ .

#### שאלה 2

ממשו את הפונקציה האיטרטיבית (לא רקורסיבית) הבאה:

```
int find_sum_three (int a[], int n, int x)
```

הפונקציה מקבלת מערך  $a$  המכיל  $n$  מספרים שלמים חיוביים (טבעיים) והממוין בסדר עולה, שכל איבריו שונים זה מזה, וכן מספר טבעי  $x$ . הפונקציה תחזיר 1 אם ניתן להציג את  $x$  כסכום של שלושה איברים שונים מהמערך  $a$ , ו-0 אם לא ניתן להציג את  $x$  כסכום של שלושה איברים שונים מהמערך  $a$ .

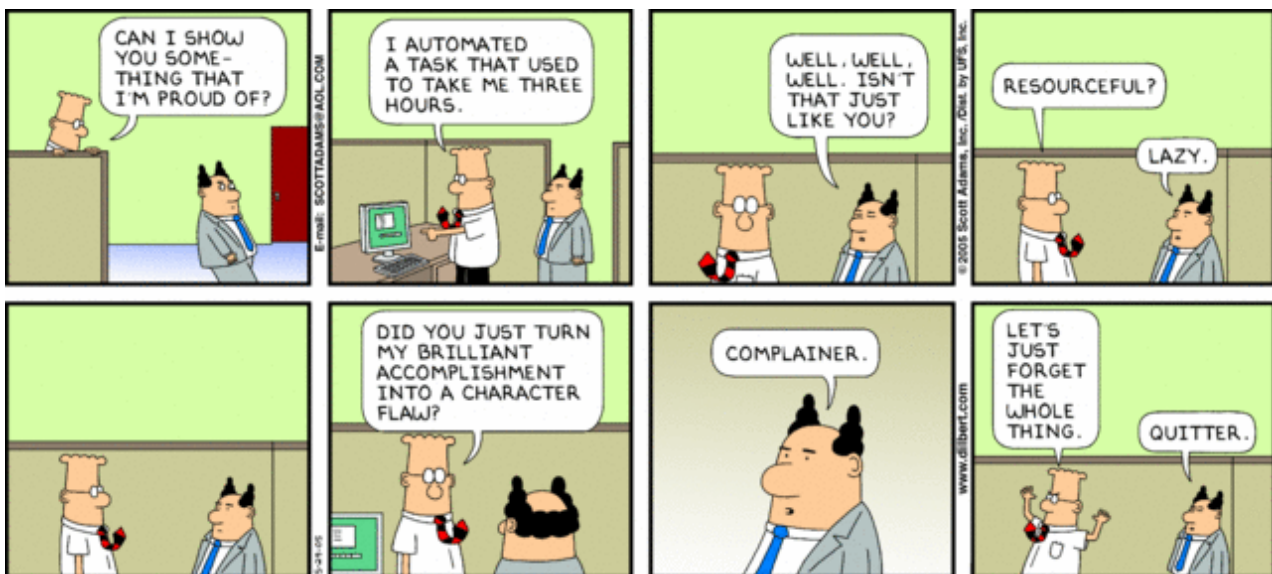
כתבו את הפונקציה  $\text{find\_sum\_three}$  כך שסיבוכיות זמן הריצה שלה תהיה  $\Theta(n^2)$ .

### שאלה 3 (רשות)

בכיתה ראינו כי ניתן להשתמש בטבלת יאנג (Young Tableau) על מנת למצוא זוג איברים במערך ממוין שסכומם שווה לערך נתון. סיבוכיות זמן הריצה של בניית הטבלה היא  $\Theta(n^2)$ , אך מרגע שהטבלה בנויה, ניתן למצוא זוג איברים בזמן  $\Theta(n)$ . פתרון אחר שראינו, לאותה בעיה אלגוריתמית, עושה שימוש בחיפוש בינארי (Binary Search), וסיבוכיותו  $\Theta(n \log n)$ .

הניחו שידוע כי ביום עבודה מתבצעים  $k$  זימונים של הפונקציה `find_sum`. מה צריך להיות ערכו של  $k$  על מנת שישתלם לנו לעשות שימוש בפתרון של טבלת יאנג, על פני הפתרון שנעזר בחיפוש בינארי?

לצורך פשטות החישוב, ניתן להתעלם מסימן ה- $\Theta$ , ולהניח כי זמן בניית הטבלה הוא בדיוק  $n^2$  פעולות, זמן ביצוע זימון יחיד של `find_sum` הנעזר בטבלת יאנג הוא בדיוק  $n$  פעולות, וזמן ביצוע זימון יחיד של `find_sum` הנעזר בחיפוש בינארי הוא בדיוק פעולות  $n \log n$ .



### שאלה 4

ממשו את יחידת הספרייה `tree.h`, עבור טיפוס הנתונים המופשט 'עץ בינארי', כפי שלמדנו בכיתה.

### שאלה 6

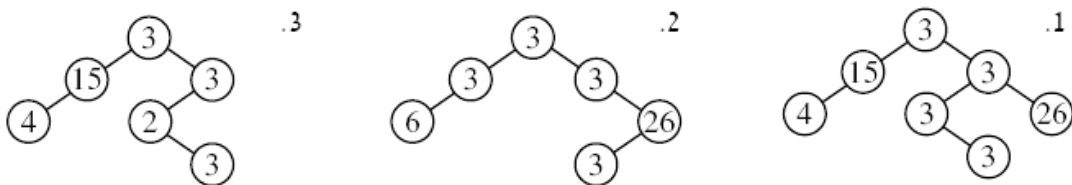
- תוצאת הסריקה התוכית (inorder) של עץ בינארי היא HEAJICBDGF, ותוצאת הסריקה הסופית (postorder) שלו היא EHJIABFGDC. שרטטו את העץ המתאים.
- האם קיים עץ בינארי לא ריק, שכל איבריו שונים, שעבורו בסריקה תחילית (preorder) ובסריקה תוכית תתקבל אותה סדרת פלט? ומה לגבי סריקה תחילית וסריקה סופית?

## שאלה 7

נתונה הפונקציה:

```
int what (tree t, int x)
{
  if (tree_empty(t))
    return 0;
  if (tree_root_retrieve(t) != x)
    return 0;
  if (tree_empty(tree_lsub(t)) && tree_empty(tree_rsub(t)))
    return 1;
  return what(tree_lsub(t),x) || what(tree_rsub(t),x);
}
```

א. איזה ערך יוחזר כשנזמן את הפונקציה כך:  $\text{what}(t,3)$  עבור כל אחד מהעצים:



ב. כתבו את טענת היציאה של הפונקציה  $\text{what}$ .



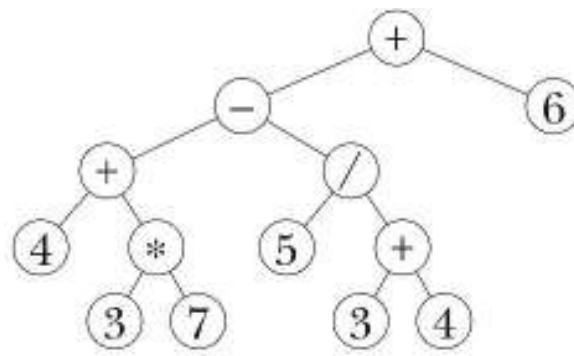
## שאלה 8

נתון הביטוי התוכי (infix) הבא:  $((A * (B / C)) - (D + E))$ . מעוניינים להמירו לביטוי תחילי (prefix) ולביטוי סופי (postfix). עשו זאת בשני שלבים: ראשית, בנו את עץ הביטוי החשבוני המתאים, על-פי האלגוריתם הרקורסיבי שנלמד בכיתה. שנית, בצעו סריקה תחילית (preorder) וסופית (postorder) של העץ, וכתבו את הביטוי המתקבל.



## שאלה 9

נתון עץ ביטוי חשבוני:



- חשבו, באמצעות האלגוריתם הרקורסיבי שנלמד בכיתה, את ערכו של הביטוי החשבוני שמיוצג על-ידי העץ.
- בצעו סריקה תחילית (preorder), תוכנית (inorder) וסופית (postorder) של העץ, ותארו את הביטוי התחילי (prefix), התוכי (infix) והסופי (postfix), המתקבלים.
- הפעילו על הביטוי הסופי שקיבלתם בסעיף ב', את האלגוריתם שלמדנו בכיתה, המשתמש במחסנית על מנת לחשב את ערכו של ביטוי סופי. האם קיבלתם את אותה התשובה כמו זו שקיבלתם בסעיף א'? במידה ולא – בדקו את עצמכם.

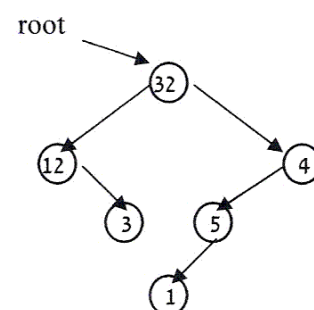
## שאלה 10

- נגדיר עץ בוסר, בתור עלה, או בתור שורש ושני תתי-עצים שכל אחד מהם הוא עץ בוסר, כאשר האיבר שבשורש תת-עץ הימני גדול מהאיבר שבשורש תת-עץ השמאלי ושניהם גדולים מהשורש.
- האם ניתן לדעת היכן נמצא הערך הגבוה ביותר בעץ? ומה לגבי הערך הנמוך ביותר? נמקו.
  - כתבו פונקציה בשפת C המקבלת מצביע לעץ בינארי, ומחזירה 1 אם זהו עץ בוסר, או 0 אם לא מדובר בעץ בוסר.
  - נסמן ב-n את מס' הצמתים בעץ. מה תהיה סיבוכיות זמן הריצה של הפונקציה שכתבתם?

## שאלה 11

נתונה הפונקציה הבאה:

```
void what (tree t, int value)
{
  if (!tree_empty(t)) {
    tree_root_modify(&t,value);
    what(tree_lsub(t),value+1);
    what(tree_rsub(t),value+1);
  }
}
```



- ונתון העץ הבא, אשר root הוא מצביע לשורשו.
- מה יתבצע בעקבות הזימון `what(root,0)`?
  - מה מטרת הפונקציה `what`?

## שאלה 12

- א. בנו יחידת הספרייה `queue.h`, עבור טיפוס הנתונים המופשט 'תור', במימוש של רשימת מקושרת, כפי שלמדנו בכיתה.
- ב. בנו תכנית בדיקה והשתמשו בה כדי לבדוק את יחידת הספרייה שבניתם בסעיף א'.
- ג. בנו יחידת הספרייה `queue.h`, עבור טיפוס הנתונים המופשט 'תור', במימוש של מערך סטטי מעגלי, כפי שלמדנו בכיתה. כדי להבחין בין תור מלא לבין תור ריק – הינכם רשאי להשתמש באחד משני הפתרונות שלמדנו (דגל בוליאני `is_empty` או מונה `count_items`).
- ד. השתמשו בתכנית הבדיקה שכתבתם בסעיף ב', כדי לבדוק את יחידת הספרייה שבניתם בסעיף ג'. האם הייתם צריכים לערוך שינויים בתכנית הבדיקה?

## שאלה 13

האלגוריתם לסריקה לפי רמות של עץ בינארי נלמד בכיתה. ממשו בסביבת העבודה את האלגוריתם. לצורך כך, עליכם להשתמש ביחידת הספרייה `tree.h` המייצגת עץ בינארי וגם ביחידת הספרייה `queue.h` המייצגת תור, אולם הפעם התור יהיה **תור של עצים בינאריים**, במקום תור של שלמים.

