

# מטלת מנחה (ממ"ן) 14

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 9-10 נושא המטלה: יעילות ורקורסיה

מספר השאלות: 4 משקל המטלה: 5 נקודות

סמסטר: 2017 מועד אחרון להגשה: 28.1.2017

השאלות במטלה זו לקוחות מבחינות גמר שונות או דומות לשאלות של בחינות גמר. חלק מהשאלות הן לתרגול בלבד ולא להגשה. אנו ממליצים מאד לענות עליהן ללא הרצה במחשב (כפי שמקובל בבחינת הגמר).

את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex14.java (בדיוק).  
את התשובות לשאלות על הסיבוכיות כתבו כחלק מה-API.

שאלה 1 - להגשה (25%)

נתונות השיטות הסטטיות הבאות:

```
private static int f (int[]a, int low, int high)
{
    int res = 0;
    for (int i=low; i<=high; i++)
        res += a[i];
    return res;
}

public static int what (int []a)
{
    int temp = 0;
    for (int i=0; i<a.length; i++)
    {
        for (int j=i; j<a.length; j++)
        {
            int c = f(a, i, j);
            if (c%3 == 0)
            {
                if (j-i+1 > temp)
                    temp = j-i+1;
            }
        }
    }
    return temp;
}
```

א. מה מבצעת השיטה what בהינתן לה מערך a מלא במספרים שלמים (חיוביים ושליילים)? הסבירו בקצרה מה מבצעת השיטה ולא כיצד היא מבצעת זאת. כלומר, כתבו מה המשמעות של המספר המוחזר מהשיטה what. (4 נק')

ב. מהי סיבוכיות הזמן של השיטה what ? (2 נק')

ג. כתבו את השיטה what כך שתבצע את מה שביצעה בסעיף א בסיבוכיות זמן ריצה קטנה יותר. (17 נק')

### שימו לב:

**השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט**

### נקודות בלבד.

ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם בסעיף ג? הסבירו תשובתכם. (2 נק')

**אל תשכחו לתעד את מה שכתבתם!**

## שאלה 2 - להגשה (25%)

נתון מערך חד-ממדי a המכיל ערכי 0 ו-1 בלבד.

לדוגמא:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1

כתבו שיטה סטטית המקבלת כפרמטר מערך a כנ"ל, ומחליפה כל ערך במערך **שאינו אפס** במרחק ממנו לנקודת האפס הקרובה ביותר (מימין או משמאל). שימו לב שב"מרחק" הכוונה פשוט למספר הצעדים במערך שיש לבצע על מנת להגיע לאיבר אפס הקרוב ביותר.

לדוגמא, עבור המערך a שלעיל, המערך a לאחר ההחלפה (כך שיכיל את המרחקים) יהיה:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	3	2	1	0	1	2	3	2	1	0	1	2

ניתן להניח שהמערך מכיל לפחות 0 אחד. כלומר, הוא אינו מכיל רק 1-ים.

חתימת השיטה היא:

```
public static void zeroDistance (int [] a)
```

**שימו לב:**

השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.

**כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם.**

**אל תשכחו לתעד את מה שכתבתם!**

### **שאלה 3- 30 נקודות (להגשה)**

נתונה מחרוזת תווים  $s$ . במהלך שליחתה ברשת האינטרנט קרתה תקלה וחלק מהתווים במחרוזת הוכפלו והופיעו ברצף מספר לא ידוע של פעמים. כלומר במחרוזת  $t$  שהתקבלה הופיעו כל התווים של  $s$ , לפי הסדר במחרוזת המקורית, אבל לפעמים תו מסוים לא הופיע פעם אחת כמו במחרוזת  $s$  אלא ברצף מספר לא ידוע של פעמים.

**נגדיר** שמחרוזת תווים  $t$  **עברה-טרנספורמציה** ממחרוזת התווים  $s$ , אם כל תו של  $s$  נמצא ב- $t$  (לפי הסדר ב- $s$ ) **לפחות** פעם אחת.

שימו לב שאם תו מופיע במחרוזת  $s$  כמה פעמים (נניח  $k$ ), אז במחרוזת  $t$  שעברה טרנספורמציה הוא יופיע לפחות  $k$  פעמים.

**לדוגמא:**

אם המחרוזת  $s = "abbed"$

אז כל המחרוזות הבאות **עברו-טרנספורמציה** ממחרוזת זו :

"abbed", "abbbced", "aabbcedd", "abbedddddd", "aaaabbed", "abbed"

וכל המחרוזות הבאות **לא עברו-טרנספורמציה** מהמחרוזת  $s$  :

"a", "abcd", "aaccbbdd"

כתבו שיטה סטטית בוליאנית רקורסיבית שחתימתה היא:

```
public static boolean isTrans (String s, String t)
```

המקבלת כפרמטרים שתי מחרוזות תווים s ו-t.

השיטה צריכה להחזיר true אם t עברה-טרנספורמציה מהמחרוזת s, ו-false אחרת.

השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.

אפשר להשתמש בהעמסת-יתר (overloading).  
אל תשכחו לתעד את מה שכתבתם!

#### שאלה 4-30 נקודות (להגשה)

כתבו שיטה בוליאנית רקורסיבית match אשר מקבלת שני מערכים. מערך אחד (a) מכיל מספרים שלמים כלשהם, ומערך שני (pattern) מכיל את המספרים 0, 1 או 2 (בלבד).

המספרים במערך התבנית pattern מייצגים כך:

המספר 1 מייצג מספר שלם בן ספרה אחת;

המספר 2 מייצג מספר שלם בן שתי ספרות;

המספר 0 מייצג מספר שלם בן ספרה אחת או שתי ספרות.

השיטה צריכה להחזיר true אם יש תת-מערך של a שמתאים לתבנית pattern, ו-false אחרת.

שימו לב שתת-המערך צריך להיות רציף (ללא "דילוגים" או "חורים")

חתימת השיטה היא: `public static boolean match (int [] a, int [] pattern)`

לדוגמא,

הקריאות הבאות בצד שמאל יחזירו את הערכים בצד ימין: (סימנו במערך a את האיברים המתאימים לתבנית pattern)

- `match ({2,3,57}, {1,0,2})` true
- `match ({5,39,67}, {1,0,2})` true
- `match ({2,3,573,4,34,35}, {1,0,2})` true
- `match ({2,3,573,4,324,35}, { })` true
- `match ({2,3}, {1,0,2})` false
- `match ({2,3,573,4,324,35}, {1,0,2})` false

אתם יכולים להניח שהמערכים מלאים במספרים לפי הכללים. בפרט במערך pattern אין מספרים אחרים חוץ מ-0, 1 או 2. אינכם צריכים לבדוק זאת.

השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.

אפשר להשתמש בהעמסת-יתר (overloading). (ואז השיטה עצמה לא חייבת להיות רקורסיבית, אבל אסור שיהיו בה לולאות).

אסור להשתמש במשתנים סטטיים (גלובליים)!  
אסור להשתמש במערך עזר.

שימו לב:

בשאלות 3 ו-4 אין צורך לדאוג ליעילות השיטה שתכתבו!

שאלה לא להגשה

לפניכם שני קטעי הקוד (שאינם קשורים זה לזה):

```
int a =3;
while (a <= n)
    a = a*a;
```

```
public void foo (int n, int m)
{
    int i = m;
    while (i > 100)
        i = i/3;
    for (int k=i ; k>=0; k--)
    {
        for (int j=1; j<n; j*=2)
            System.out.print(k + "\t" + j);
        System.out.println();
    }
}
```

מה סיבוכיות זמן הריצה של קטעי הקוד האלו?

להזכירכם – חוקי הלוגריתמים:

$$\log_a m \times n = \log_a m + \log_a n$$

$$\log_a \frac{m}{n} = \log_a m - \log_a n$$

$$\log_a n^m = m \times \log_a n$$

## שאלה לא להגשה

התבוננו בשיטות הבאות:

```
public static void f(int [][] a,
                    int a1, int b1, int a2, int b2)
{
    int temp = a[a1][b1] ;
    a[a1][b1] = a[a2][b2] ;
    a[a2][b2] = temp ;
    if (b1 < a[0].length-1)
        f(a, a1, b1+1, a2, b2-1) ;
    else if (a1+1 < a2-1)
        f(a, a1+1, 0, a2-1, a[0].length-1) ;
}

public static void printArray(int[][] a)
{
    for (int i= 0; i< a.length; i++)
    {
        for (int j= 0; j< a[i].length; j++)
            System.out.print (a[i][j] + "\t");
        System.out.println();
    }
}
```

נניח שנתונה השיטה main הבאה:

```
public static void main (String [] args)
{
    int[][] arr = {{1, 2, 3, 4}, {5, 6, 7, 8}} ;
    f(arr, 0, 0, arr.length-1, arr[0].length-1) ;
    printArray (arr);
}
```

1. מה הפלט שתפיק השיטה main?

2. כמה קריאות רקורסיביות מתבצעות בזימון

```
f(arr, 0, 0, arr.length-1, arr[0].length-1) ;
```

## שאלה לא להגשה

לפניכם קטע הקוד הבא:

```
public static int foo (int a, int b)
{
    if (a>3)
        return 2 + foo (b-1, a+1);
    if (b<=4)
        return 1 + foo (a-1, b+1);
    return 0;
}
```

לכל אחת מהקריאות הבאות לשיטה foo, ענו אם היא תעצור, ואם כן, מה היא תחזיר.

א. foo (3, 4)

ב. foo (4, 5)

## הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות השיטות יהיו **בדיוק** כפי שמוגדר בממ"ן.
3. עליכם לתעד את כל השיטות שאתם כותבים בתיעוד API ובתיעוד פנימי המסביר מה עשיתם בשיטה. בתיעוד זה כתבו גם מה הסיבוכיות של השיטות (בשאלות 1 ו-2).
4. את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex14.java (**בדיוק**). ארוזו אותו בתוך קובץ zip. אין לשלוח קבצים נוספים.

## בהצלחה

