

# 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: 17.6.2017

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 – 25 נקודות (להגשה)

נתאר את בעיית מציאת "בור" במערך דו-ממדי ריבועי:

קלט: מערך דו-ממדי ריבועי בגודל  $n \times n$  המלא באפסים ואחדים בלבד.

נגדיר ש- $k$  הוא בור (sink) אם בשורה ה- $k$  ית כל הערכים הם 0, ובעמודה ה- $k$  ית כל

הערכים הם 1 (חוץ מהאיבר  $[k][k]$  עצמו שהוא 0).

פלט: האם קיים מספר  $k$  המהווה בור במערך? אם כן, יש להחזיר את ערכו אחרת יש להחזיר -1.

לדוגמא: במערך A הוא "בור", ובמערך B אין בור.

B						A					
0	1	0	0	0	1	0	1	0	1	1	0
1	0	0	1	1	1	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0	1	0	1
1	1	1	1	1	1	0	0	0	0	0	0
0	1	0	1	0	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1	0	1	1	1

כתבו שיטה יעילה הפותרת את הבעיה. השיטה תחזיר את המספר  $k$  המהווה בור במערך, אם קיים אחד כזה, ו-1 אם לא קיים בור במערך. כתבו והסבירו מה סיבוכיות השיטה שכתבתם.

חתימת השיטה תהיה :

```
public static int isSink (int [][] mat)
```

שימו לב,

השיטה צריכה להיות יעילה ככל הניתן.

שיטה שתעבוד בסיבוכיות גבוהה מזו הנדרשת (במקום או בזמן) לא תקבל את מירב הנקודות. פתרון נכון שיהיה בסיבוכיות  $O(n^2)$ , יזכה את כותבו ב- 10 נקודות בלבד.

אל תשכחו לתעד את מה שכתבתם!

## שאלה 2 - 25 נקודות (להגשה)

נתונה מטריצה  $mat$  מגודל  $n \times n$  המיוצגת על ידי מערך דו מימדי. המטריצה מכילה מספרים שלמים. אנו מניחים כי  $n = 2^k$ , עבור  $k$  טבעי כלשהו. נאמר כי המטריצה ממוינת אם בחלוקה של המטריצה ל- 4 רבעים (לפי הציר), כל איבר ברביע הראשון קטן או שווה לכל איבר ברביע השני, כל איבר ברביע השני קטן או שווה לכל איבר ברביע השלישי, וכל איבר ברביע השלישי קטן או שווה מכל איבר ברביעי. כמו להלן:

1 2

3 4

כמו כן התכונה הנ"ל מתקיימת רקורסיבית על כל אחד מהרבעים.

למשל, המטריצה הבאה ממוינת:

-4	-2	5	9
2	5	12	13
13	20	25	25
22	24	49	57

כתבו שיטה עם החתימה הבאה:

```
public static boolean find(int [][] mat, int x)
```

המחזירה true אם ורק אם הערך  $x$  נמצא במטריצה הממוינת  $mat$  (נתון שהמטריצה אכן ממוינת ואין צורך לבדוק זאת).

השיטה צריכה להיות יעילה ככל הניתן.

פתרון נכון שאינו מספיק יעיל יקבל ניקוד חלקי בלבד. פתרון מסדר גודל ריבועי  $O(n^2)$  לא יתקבל בכלל.

אין להקצות זיכרון נוסף בגודל שאינו קבוע.  
ניתן להשתמש בשיטות עזר ככל הנדרש. בחישוב הסיבוכיות צריך לחשב גם את הזמן והמקום של שיטות העזר.  
שימו לב, השיטה לא חייבת להיות רקורסיבית!

כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם.

אל תשכחו לתעד את מה שכתבתם!

### שאלה 3-25 נקודות (להגשה)

תהי נתונה קבוצה  $S$  של מספרים טבעיים (שלמים גדולים מ-0), ויהי  $n$  מספר טבעי כלשהו. נאמר ש- $n$  הוא סכום מתוך  $S$  אם ניתן לבטא את  $S$  כסכום של מספרים (עם או בלי חזרות) מתוך  $S$ .

דוגמאות:

- תהי  $S = \{4, 5\}$ , אזי למשל 13 הוא סכום מתוך  $S$  שכן  $4 + 4 + 5 = 13$ , אבל 6 אינו סכום מתוך  $S$ , שכן לא ניתן לבטא את 6 כסכום של מספרים מתוך  $S$ .
- תהי  $S = \{4, 9, 3\}$ , אזי למשל 15 הוא סכום מתוך  $S$  שכן  $9 + 3 + 3 = 15$ , אבל 5 אינו סכום מתוך  $S$ , שכן לא ניתן לבטא את 5 כסכום של מספרים מתוך  $S$ .

שימו לב: 0 הוא סכום מתוך כל קבוצה  $S$ , שכן ניתן לבטא את 0 כסכום של קבוצה ריקה של מספרים. כמו כן, לכל מספר טבעי  $n$  השייך לקבוצה  $S$ ,  $n$  הוא סכום מתוך  $S$ , כן ניתן לבטא את  $n$  כסכום של עצמו בלבד.

כתבו שיטה סטטית רקורסיבית בוליאנית המקבלת כפרמטרים מערך  $s$  מלא במספרים טבעיים שונים זה מזה, המהווים את איברי הקבוצה, ומספר טבעי  $n$ , ומחזירה true אם  $n$  הוא סכום מתוך  $s$ . אחרת, השיטה תחזיר false.  
המערך לא ממוין.

חתימת השיטה היא:

```
public static boolean isSumOf(int [] s, int n)
```

אתם יכולים להניח כי הפרמטרים תקינים. המערך  $s$  אינו null ואינו ריק, וכל האיברים שבו הם טבעיים ושונים זה מזה, והמספר  $n$  הוא טבעי.

השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. ניתן להשתמש בשיטות עזר, אך גם הן לא יכולות להכיל לולאות.

אפשר להשתמש בהעמסת-יתר (overloading).

אל תשכחו לתעד את מה שכתבתם!

#### שאלה 4-25 נקודות (להגשה)

נתון מערך דו-ממדי ריבועי המכיל בתוכו מספרים שלמים. נגדיר מסלול (path) במערך המתחיל בתא  $[x1][y1]$  ומסתיים בתא  $[x2][y2]$  כך: סדרה של תאים במערך באופן שמתקיים כי:

- התא הראשון בסדרה הוא  $[x1][y1]$
- התא האחרון בסדרה הוא  $[x2][y2]$
- המעבר מתא הוא רק לשכניו כאשר שכניו של תא הם התאים הצמודים אליו שמעליו, מתחתיו, מימינו ומשמאלו. במקרה ורק חלק מהשכנים קיימים (כאשר התא  $[i][j]$  נמצא בשולי המערך), יש להתחשב רק בשכנים הקיימים.
- כל תא במערך מופיע לכל היותר פעם יחידה במסלול.

**לדוגמא**, במערך בן חמש שורות וחמש עמודות קיימים המסלולים הבאים בין התא  $[0][1]$  לתא  $[2][4]$ :

- $[0][1] - [0][2] - [0][3] - [1][3] - [1][4] - [2][4]$
  - $[0][1] - [1][1] - [2][1] - [2][0] - [3][0] - [3][1] - [3][2] - [3][3] - [3][4] - [2][4]$
- וקיימים, כמובן, עוד מסלולים רבים.

כתבו שיטה סטטית **רקורסיבית** המקבלת כפרמטר מערך דו-ממדי `mat` מלא במספרים שלמים חיוביים, וזוג תאים במערך המצויים מעל לאלכסון הראשי (כלומר, האלכסון `mat[i][i]`). השיטה צריכה להחזיר את מספר המסלולים שאינם חוצים את האלכסון הראשי (אך הם עשויים להכיל תאים המצויים באלכסון זה) הקיימים בין שני התאים. **בדוגמא לעיל**, המסלול הראשון אינו חוצה את האלכסון הראשי, אך המסלול השני כן חוצה, ולכן הוא לא ייספר.

חתימת השיטה:

```
public static int numPaths (int[] [] mat,
                           int x1, int y1, int x2, int y2)
```

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).

אסור להשתמש במשתנים סטטיים (גלובליים)!

אסור להשתמש במערך עזר.

ערכי המטריצה לאחר הרצת השיטה numPaths צריכים להיות תואמים לערכי המטריצה לפני הרצת השיטה.

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

