

מטלת מנחה (ממ"ן) 14

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 14-16 נושא המטלה: רקורסיה ויעילות

מספר השאלות: 4 משקל המטלה: 5 נקודות

סמסטר: 2015 מועד אחרון להגשה: 17.1.2015

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

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 ושליליים)? 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 - להגשה (25%)

סעיף א - 8 נקודות

בניו יורק יש בניין בן n קומות. ספיידרמן רוצה לטפס על הבניין ולהגיע מהרחוב אל הקומה האחרונה. הוא יכול לטפס בכל צעד קומה אחת או שתיים. כתבו שיטה סטטית רקורסיבית `int spiderman(int n)` אשר מחשבת בכמה דרכים שונות יכול ספיידרמן להגיע לקומה האחרונה של הבניין. **לדוגמא**, קריאה לשיטה `spiderman(4)` תחזיר ערך 5 מכיוון שבבניין בעל 4 קומות יכול ספיידרמן להגיע לקומה האחרונה ב-5 דרכים שונות:

1. לטפס קומה אחת בכל פעם עד שיגיע לקומה הרביעית
2. לטפס לקומה ראשונה, אח"כ לקומה שנייה ובצעד השלישי לדלג שתי קומות לקומה הרביעית.
3. לטפס לקומה ראשונה, אח"כ לשלישית ובצעד השלישי לרביעית.
4. לטפס לקומה השנייה, אח"כ לשלישית ובצעד השלישי לרביעית.
5. בצעד הראשון לטפס לקומה השנייה ובצעד השני לטפס לרביעית.

סעיף ב - 17 נקודות

בבניין נתון יש n קומות ($n < 30$) ובקומה ה-20 יש תאי טלפון בהם יכול ספיידרמן להוריד את התחפושת ולהפוך לפיטר פרקר ומשם לקחת את המעלית לקומה האחרונה. כתבו שיטה סטטית רקורסיבית בשם `int spidermanPhoneBooth20(int n)` אשר מחשבת בכמה דרכים שונות יכול ספיידרמן להגיע לקומה האחרונה (קומה n) בבניין שבו בקומה ה-20 יש מעלית ישירה לקומה האחרונה. שימו לב, כעת יש לספיידרמן קיצור דרך שהוא **חייב** להשתמש בו, כאשר הוא מגיע לקומה זו (הוא לא יכול להגיע לקומה 20 ולהמשיך ממנה בלי לקחת את המעלית). אתם רשאים להשתמש בקריאות לשיטה מסעיף א.

לדוגמא, הנה שלוש דרכים שונות (מתוך X האפשרויות) לטפס על בניין עם 26 קומות (ומעלית בקומה ה-20):

1. לטפס קומה אחת בכל פעם עד שיגיע לקומה ה-20 ומשם ישירות לקומה האחרונה.
2. לטפס קומה אחת בכל פעם עד שיגיע לקומה ה-19, אח"כ לטפס לקומה 21 ומשם קומה אחת בכל פעם עד שיגיע לקומה האחרונה.
3. לטפס לקומה ראשונה, אח"כ לדלג שתי קומות בכל פעם (לקומה השלישית, החמישית, השביעית...) עד שיגיע לקומה ה-25 ומשם לטפס לקומה האחרונה.

דוגמה לטיפוס לא אפשרי:

- לטפס קומה אחת בכל פעם עד שיגיע לקומה האחרונה.
- טיפוס זה אינו אפשרי מכיוון שהוא לא יכול להגיע לקומה 21 ולהמשיך ממנה בלי לקחת את המעלית לקומה האחרונה.

אין להשתמש בלולאות ו/או במערך בשאלה זו.
שימו לב שהשיטה 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)!

שאלה 4 - להגשה (25%)

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:

- א. התא הראשון בסדרה הוא $[x1][y1]$
- ב. התא האחרון בסדרה הוא $[x2][y2]$
- ג. 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.
- ד. 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.

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**). ארוז אותו בתוך קובץ zip. אין לשלוח קבצים נוספים.

בהצלחה