

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: 2020 מועד אחרון להגשה: 20.6.2020

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)

אמיר ותמר משחקים במשחק. על לוח המשחק מונחים K מטבעות בשורה. לכל מטבע בשורה יש ערך שהוא מספר חיובי (ממש, גדול מ-0), והשחקנים יודעים בתחילת המשחק את ערכיהם של כל המטבעות.



המשחק מתנהל בתורות, לסירוגין. בכל תור, השחקן שמשחק בוחר מטבע מאחד משני קצוות השורה ולוקח אותו לקופה שלו. לאחר K תורות נגמרים המטבעות בשורה. בשלב זה סופרים את סכום ערכי המטבעות שבקופה של כל אחד מהשחקנים. השחקן שצבר סכום גדול יותר, מנצח במשחק. במקרה של שוויון בסכומים, המשחק מוכרז כתיקו.

אמיר משחק ראשון. הוא ממש לא רוצה להפסיד. לא אכפת לו אם המשחק יסתיים בתיקו או בניצחון שלו. מצאו אסטרטגיה עבור אמיר שתבטיח שהוא לא יפסיד במשחק. כמוכן שאמיר יכול לחשב מראש את עץ המהלכים המלא של המשחק: איך תמר יכולה להגיב לכל מהלך שלו, ואז איך הוא יגיב לכל מהלך שלה, וכו'. הבעיה בפתרון כזה היא שהעץ הזה עצום: מספר המשחקים השונים שאמיר ותמר יכולים לשחק הוא 2^K ואם K גדול, גם למחשב החזק ביותר בעולם, החישוב הזה ייקח טריליוני שנים. לכן, אנחנו מעוניינים בפתרון יעיל לבעיה – כזה שדורש מאמיר לבצע מעט פעולות כדי לחשב לעצמו אסטרטגיה.

כמוכן ש- K (מספר המטבעות) הוא זוגי, כדי שלא יהיה יתרון במספר המטבעות לאחד מהצדדים.

נניח שמערך חד-ממדי arr מכיל את שורת המטבעות.

כתבו שיטה סטטית בשם win המחשבת לאמיר מהלך אפשרי של המשחק כך שהוא בהכרח לא מפסיד בו.

השיטה מקבלת את מערך המטבעות כפרמטר, ומדפיסה את בחירות השחקנים בכל שלב במשחק. בסוף התהליך יודפס מה היה הסכום הכולל של אמיר ומה הסכום הכולל של תמר.

עליכם להניח במהלך השיטה שכאשר תורה של תמר, היא תמיד תיקח את הגדול מבין שני המטבעות שבקצוות.

חתימת השיטה היא :

```
public static void win (int [] arr)
```

לדוגמא, אם רשימת המטבעות שבמערך היא :

0	1	2	3	4	5	6	7
15	19	21	13	14	30	23	16

פלט אפשרי למשחק יהיה :

Amir took 16

Tamar took 23

Amir took 30

Tamar took 15

Amir took 19

Tamar took 21

Amir took 13

Tamar took 14

Final Score:

Amir total 78

Tamar total 73

שימו לב:

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

ניתן להשתמש בשיטות עזר ככל הנדרש. בחישוב הסיבוכיות צריך לחשב גם את הזמן והמקום של שיטות העזר. **פורמט הפלט של השיטה צריך להיות בדיוק לפי דוגמת הפלט הנתונה.**

אל תשכחו לתעד את השיטה שכתבתם.

כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם. הסבירו את תשובתכם.

שאלה 2 - 25 נקודות (להגשה)

כתבו שיטה סטטית המקבלת כפרמטר מערך `arr` המלא במספרים שלמים, (חיוביים ושליילים). השיטה צריכה למצוא שלשה של מספרים במערך (לאו דווקא סמוכים זה לזה) שהמכפלה שלהם היא המקסימלית. השיטה תדפיס את שלושת המספרים האלו ותחזיר את המכפלה המקסימלית. **אין חשיבות לסדר ההדפסה.**
שימו לב – עליכם להדפיס את שלושת המספרים מופרדים ברווח ללא טקסט נוסף.
אפשר להניח שאורך המערך הוא לפחות 3.

לדוגמא:

- אם המערך `arr` מכיל את המספרים הבאים:

0	1	2	3	4
-4	1	-8	9	6

אזי תודפס השלשה: 9 -8 -4
ותוחזר המכפלה $-4 * -8 * 9 = 288$

חתימת השיטה היא:

```
public static int findTriplet (int [] arr)
```

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.

מה סיבוכיות זמן הריצה של השיטה שכתבתם? הסבירו תשובתכם.

אל תשכחו לתעד את מה שכתבתם!

שאלה 3- 25 נקודות (להגשה)

נתונות מחרוזת תווים str ומחרוזת תווים pattern .

כתבו שיטה סטטית רקורסיבית, המחזירה את מספר הפעמים שהמחרוזת pattern מופיעה בתוך המחרוזת str כתת-סדרה. שימו לב שהמחרוזת pattern לא צריכה להופיע ברציפות בתוך str.

לדוגמא,

אם המחרוזת str = "subsequence" והמחרוזת pattern = "sue" אז השיטה תחזיר 7 שכן המחרוזת "sue" נמצאת 7 פעמים כתת-סדרה במחרוזת "sequence" כמפורט להלן:

1. subsequence
2. subsequence
3. subsequence
4. subsequence
5. subsequence
6. subsequence
7. subsequence

חתימת השיטה היא:

```
public static int count (String str, String pattern)
```

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).

אין צורך לדאוג ליעילות השיטה, אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות! אל תשכחו לתעד את מה שכתבתם!

שימו לב, **בפתרון שני-הסעיפים** מתוך השיטות המוגדרות במחלקה String, מותר להשתמש אך ורק בשיטות שלהלן **בפתרון השאלה**:

- `public char charAt(int i)` - המחזירה את התו במקום ה-i במחרוזת (עליה היא מופעלת).

- `public int length()` - המחזירה את אורך המחרוזת עליה היא מופעלת

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

שאלה 4- 25 נקודות (להגשה)

הנסיך מקפץ על גגות העיר. מטרתו למצוא את הרשע ולהילחם בו, על מנת להציל את הנסיכה. מפת הגבהים של גגות העיר מיוצגת על-ידי מערך דו-ממדי ריבועי המכיל מספרים שלמים. הניחו כי אם התא מכיל מספר אי-שלילי, הוא מייצג גובה של גג. המיקום של הרשע מסומן על-ידי המספר 1-.

בכל צעד הנסיך יכול להתקדם למשבצת סמוכה: צפונה, דרומה, מזרחה או מערבה (לא באלכסון). אם המשבצת הסמוכה נמצאת באותו גובה – יכול הנסיך ללכת אליה. בנוסף, יכול הנסיך לטפס על גג בגובה יחידה אחת, או לרדת מגג בגובה יחידה אחת או שתיים. אם הוא מנסה לטפס לגובה של יותר מיחידה אחת או לרדת לגובה של יותר משתי יחידות, הוא נפסל מיד. כשהנסיך נמצא על גג סמוך לרשע (אחד מארבעת שכניו) הוא יכול לקפוץ אליו ללא קשר להפרש הגבהים בינו לבין הרשע.

עליכם לכתוב שיטה סטטית רקורסיבית שתתכן את המסלול שיביא את הנסיך לרשע במספר המשבצות הנמוך ביותר בלי להיפסל. אפשר להניח כי המערך מייצג בצורה נכונה את הגבהים של גגות העיר. כל הערכים בתאים הם מספרים אי-שליליים ויש רק תא אחד שמחזיק מספר שלילי והוא 1-1. אין תאים נוספים במערך שיש בהם מספרים שליליים.

חתימת השיטה היא:

```
public static int prince(int[][] drm, int i, int j)
```

כאשר מפת הגבהים של גגות העיר נתונה על-ידי הפרמטר `drm` (Digital Roof Map) ואילו `i` ו-`j` מציינים את אינדקס השורה והעמודה בהתאמה של התא בו מתחיל הנסיך. על השיטה להחזיר את מספר התאים שעל הנסיך לעבור במסלול הקצר ביותר או 1- אם אין מסלול חוקי כזה. לאחר ריצת השיטה על המפה (המערך) להישאר ללא שינוי.

לדוגמא, עבור המפה הבאה:

	0	1	2	3	4
0	2	0	1	2	3
1	2	3	5	5	4
2	8	-1	6	8	7
3	3	4	7	2	4
4	2	4	3	1	2

הרשע נמצא בתא בשורה 2 בעמודה 1.

אם הנסיך נמצא בתא (0,0) קיימים שלושה מסלולים המובילים אותו לרשע. הם מסומנים במפות להלן:

2	0	1	2	3
2	3	5	5	4
8	-1	6	8	7
3	4	7	2	4
2	4	3	1	2

מסלול באורך 10

2	0	1	2	3
2	3	5	5	4
8	-1	6	8	7
3	4	7	2	4
2	4	3	1	2

מסלול באורך 10

2	0	1	2	3
2	3	5	5	4
8	-1	6	8	7
3	4	7	2	4
2	4	3	1	2

מסלול באורך 4

לכן השיטה תחזיר 4.

באותה מפה, אם הנסיך נמצא בתא (4,4) אין מסלול חוקי בו הוא יכול ללכת עד הרשע ולכן השיטה תחזיר -1.

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).

מותר לשנות את המערך drn במהלך השיטה, אבל חובה להחזיר אותו למצבו ההתחלתי בסופה.

אין צורך לדאוג ליעילות השיטה, אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!

אל תשכחו לתעד את מה שכתבתם.

שימו לב:

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

בכל השאלות - אל תשכחו לתעד את מה שכתבתם!

שימו לב ששמנו טסטר באתר הקורס. חובה שטסטר ירוץ ללא שגיאות קומפילציה עם המחלקה שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטר ירוצו עם המחלקות ללא שגיאות קומפילציה. אם הטסטר לא ירוץ ללא שגיאות קומפילציה הציון במטלה יהיה אפס.

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

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

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)

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) ;
```

