

מטלת מנחה (ממ"ן) 12

הקורס: 20454 - מבוא למדעי המחשב ושפת Java ב

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

מספר השאלות: 2 משקל המטלה: 5 נקודות

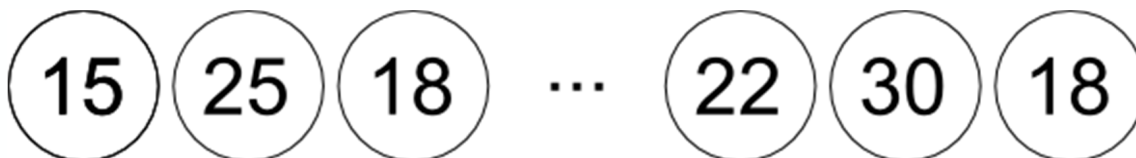
סמסטר: 2020 מועד אחרון להגשה: 9.5.2020

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

את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex12.java (בדיוק).
את התשובות לשאלות על הסיבוכיות כתבו כחלק מה-API.

שאלה 1 – 50 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.
כמובן 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.

כמובן 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.

אל תשכחו לתעד את השיטה שכתבתם.

כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם.

שאלה 2 - 50 נקודות (להגשה)

כתבו שיטה סטטית המקבלת כפרמטר מערך `arr` המלא במספרים שלמים, (חיוביים ושלייליים). 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.

לדוגמא:

• אם המערך `arr` מכיל את המספרים הבאים:

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

אזי תודפס השלשה: $-4 \quad -8 \quad 9$
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!

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 שהטסטרים 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.
4. את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex12.java (**בדיוק**). ארזו 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?

שאלה לא להגשה

לפניכם קטע הקוד הבא:

```
for (int i=1; i<n; i++)
{
    for (int j=i; j<=m; j++)
        S1;
    for (int k=t; k>=i; k--)
        S2;
}
```

כאשר S1 ו-S2 הם משפטים פשוטים והזמן הדרוש לביצועם הוא $O(1)$. כמו כן, S1 ו-S2 אינם

משנים את הערכים של i, j, k, n, m, t .

מה סיבוכיות זמן הריצה של קטע הקוד הנתון? הקיפו בעיגול את התשובה הנכונה.

1. $O(\max(n, m, t))$

2. $O(n*m*t)$

3. $O(n+m*t)$

4. $O(n*(m+t))$

שאלה לא להגשה

עבור כל אחת מהפעולות שבטבלה, ציינו מהו **סדר גודל זמן הריצה** הדרוש כדי לבצע את הפעולה במערך ממוין ובמערך שאינו ממוין.

כלומר, עליכם למצוא אלגוריתם יעיל ככל האפשר ולכתוב מה סדר גודל זמן הריצה שלו. כל זאת ללא שימוש במערכי עזר, ובלי לשנות את אופי המערכים (ממוין/לא ממוין).

גודל המערך הוא n .

בשני המקרים ניתן להניח ששום פעולה לא תגרום לחריגה מגבולות המערך.

הפעולה	זמן במערך ממוין	זמן במערך לא ממוין
חישוב ממוצע האיברים		
הכנסת איבר למערך		
חישוב סכום האיבר השלישי בגודלו עם האיבר השביעי בגודלו		
בדיקה האם יש שני איברים שונים במערך		