

הפקולטה למדעי הטבע - המחלקה למדעי המחשב

סמסטר א', מועד א', יום רביעי י"ז שבט תשע"ו, Wednesday, January 27, 2016

מחלקה: מדעי המחשב

מרצה: פרופ' ואדים (דוד) לויט, גב' אליזבת איצקוביץ

שם הקורס: אלגוריתמים 1

מס' הקורס: 7027010

משך הבחינה: 3 שעות

חל איסור על שימוש בכל חומר עזר.

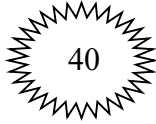
הנחיות כלליות:

- המבחן ייבדק בצורה אוטומטית ע"י תוכנת מחשב שתשתמש בשמות המוזכרים להלן.
- יש לרשום אלגוריתמים, סיבוכיות והוכחות בקובץ **word** בשם **algorithms**.
- יש לדחוס את קובץ **word** וכל קבצי **java** בקובץ אחד. שם הקובץ – מספר תעודת זהות, סוג הקובץ - **rar** או **zip**.
- שפת תכנות – **java**.

הנחיות לתכנות:

- יש לבנות **java project** בשם **Exam**
- לפתרון של כל שאלה צריך לבנות מחלקה נפרדת.
- את קובץ הפרויקט ואת השאלון ניתן להוריד מ"מתזמן מבחנים".

המבחן שלא יעמוד בדרשות אלו לא ייבדק!



בעיה מס' 1 : נתון בית בעל n קומות ושני כדורי זכוכית זהים.

גבהים של הקומות מיוצגים על ידי הסדרה $x_1 \leq x_2 \leq \dots \leq x_n$, כאשר x_i מהווה פוטנציאל של קומה מספר i לגרום לשבירת הכדור.

הכדור מיוצג ע"י מספר a שמהווה חוזק שבירת הכדור.

הכדור יישבר בנפילתו מקומה מספר i אם ורק אם $a < x_i$.

יש ליישם אלגוריתם יעיל המוצא את הקומה הנמוכה ביותר שבנפילה ממנה הכדור יישבר.

יש להכין מחלקה בשם **Q1**.

בתוך המחלקה יש לכתוב בנאי המחלקה שמקבל מערך של גבהי הקומות x וחוזק שבירת הכדור a :

```
public Q1(int[] x, int a){...}
```

ושלוש הפונקציות הבאות:

(א) פונקציה שמחשבת ומחזירה את מספר הפעמים שהכדור נשבר, במילים אחרות, את מספר הפעמים בהן מתקיים אי השוויון $a < x_i$:

```
public int numberOfBreaks(){...}
```

(ב) פונקציה שמחשבת ומחזירה מספר השוואות בין a לבין איברי המערך $x_1 \leq x_2 \leq \dots \leq x_n$:

```
public int numberOfChecking(){...}
```

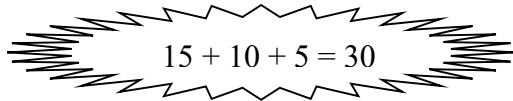
(ג) פונקציה שמחשבת ומחזירה את אינדקס הקומה שבנפילה ממנה הכדור נשבר:

```
public int floorIndex(){...}
```

אלגוריתם, סיבוכיות, דוגמא והוכחות.

הפקולטה למדעי הטבע - המחלקה למדעי המחשב

בעיה מס' 2: נתונה סדרת מספרים $a_1, a_2, \dots, a_{n-1}, a_n$. תת-סדרה של סדרה זו



$$15 + 10 + 5 = 30$$

נקראת סדרה **עולה-יורדת** (bitonic) כאשר היא עולה באופן מונוטוני ולאחר מכן יורדת באופן מונוטוני. כאשר סדרה ממוינת בסדר עולה היא נחשבת כסדרה עולה-יורדת עם חלק יורד ריק. באופן דומה סדרה שממוינת בסדר יורד נחשבת כסדרה עולה-יורדת עם חלק עולה ריק.

דוגמה 1: נתונה סדרה 1, 11, 2, 10, 4, 5, 2, 1.

תת-סדרה עולה-יורדת ארוכה ביותר היא 1, 2, 10, 5, 2, 1

דוגמה 2: תת-סדרה עולה-יורדת ארוכה ביותר של הסדרה 12, 11, 40, 5, 3, 1 היא 12, 11, 5, 3, 1.

דוגמה 3: תת-סדרה עולה-יורדת ארוכה ביותר של סדרה 1, 2, 10, 20, 32, 40, 43

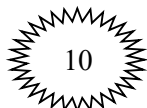
יש להכין מחלקה בשם **Q2**. בתוך המחלקה יש לכתוב שלוש פונקציות סטטיות הבאות:

(א) פונקציה סטטית המקבלת מערך של מספרים שלמים ומחזירה אורכו של תת-מערך **עולה** ארוך ביותר בסיבוכיות $O(n \cdot \log n)$.



```
public static int lis(int[] arr){...}
```

(ב) פונקציה סטטית המקבלת מערך של מספרים שלמים ומחזירה אורכו של תת-מערך **יורד** ארוך ביותר בסיבוכיות $O(n \cdot \log n)$.



```
public static int lds(int[] arr){...}
```

(ג) פונקציה סטטית המקבלת מערך של מספרים שלמים ומחזירה אורכו של תת-מערך **עולה-יורד** ארוך ביותר בסיבוכיות $O(n \cdot \log n)$.



```
public static int lbs(int[] arr){...}
```

אלגוריתם, סיבוכיות, דוגמא והוכחות.

הפקולטה למדעי הטבע - המחלקה למדעי המחשב

$$15 + 15 = 30$$

בעיה מס' 3: יישמו את האלגוריתם המחשב את עלות המסלול הזול ביותר ואת מספר המסלולים הזולים ביותר

בין שני קודקודים מסוימים בדף חשבוני בגודל M על N (עם משקלים מונחים על הצלעות). בנוסף, בדף זה ישנו אזור הנקרא "שטח מת", שאף מסלול לא יכול לעבור בו. צורתו של השטח הוא מלבן שצלעותיו מקבילות לצירים X ו- Y והוא מוגדר ע"י שתי נקודות: נקודה שמאלית תחתונה ונקודה ימנית עליונה.

קלט: מערך דו-ממדי של קודקודים. כל קודקוד (Node) מכיל את המשקלים של שתי הצלעות היוצאות ממנו לכיוונים ימין ומעלה:

x - משקל המונח על הצלע האופקית, (הכיוון, כמו בציר ה- X , ימין).

y - משקל המונח על הצלע האנכית, (הכיוון כלפי מעלה, כמו בציר ה- Y).

p_1 - נקודה שמאלית תחתונה של "השטח המת", p_2 - נקודה ימנית עליונה של "השטח המת".



פלט:

(א) עלות המסלול הזול ביותר,

(ב) מספר המסלולים הזולים ביותר.



דוגמה:

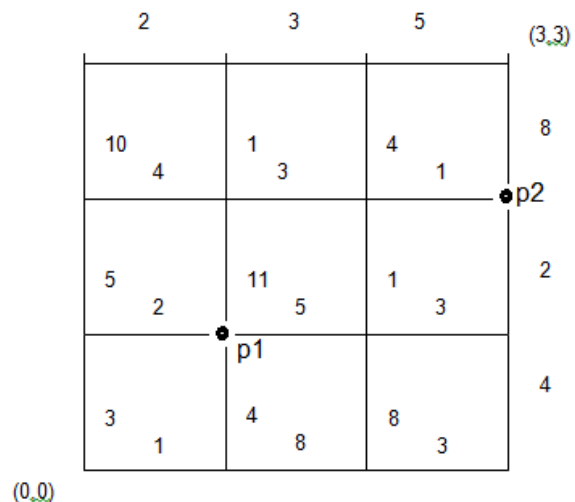
קלט: במערך דו-ממדי שלהלן:

קודקוד $(0,0)$ נראה כך: $x = 1; y = 3;$

קודקוד $(0,3)$ נראה כך: $x = 0; y = 4;$

קודקוד $(3,3)$ נראה כך: $x = 0; y = 0;$

$p_1=(1,1), p_2=(2,3)$



פלט: עלות המסלול הזול ביותר 28, מספר המסלולים הזולים ביותר 1.

הפקולטה למדעי הטבע - המחלקה למדעי המחשב

יש להכין מחלקה בשם Q3.
יש לכתוב בנאי המחלקה ושתי שיטות.

כמובן, ניתן לכתוב שיטות עזר לפי הצורך.

1. בנאי המחלקה מקבל משקלים כמערך דו-ממדי של קדקודים ושתי נקודות:

```
public Q3(Node[][] nodes, Point p1, Point p2){...}
```

מבנה של קדקוד אחד מיוצג ע"י מחלקת Node:

```
class Node{
    int x, y
    public Node(int x, int y){
        this.x = x;
        this.y = y;
    }
}
```

2. שיטה שמחזירה את עלות המסלול הזול ביותר:

```
public int cheapestPathPrice(){...}
```

3. שיטה שמחזירה מספר המסלולים הזולים ביותר:

```
public int numOfCheapestPaths(){...}
```

הקוד שמבצע מילוי המערך דו-ממדי שבדוגמה ומחלקת Point נמצאים בעמוד הבא.

אלגוריתם, סיבוכיות, דוגמא והוכחות.

הפקולטה למדעי הטבע - המחלקה למדעי המחשב

```
public static Node[][] initMatOfNodes(){ // n = 4
    int n=4;
    Node mat[][] = new Node[n][n];
    //the 1-st row
    mat[0][0] = new Node(1,3);
    mat[0][1] = new Node(8,4);
    mat[0][2] = new Node(3,8);
    mat[0][3] = new Node(0,4);
    //the 2-nd row
    mat[1][0] = new Node(2,5);
    mat[1][1] = new Node(5,11);
    mat[1][2] = new Node(3,1);
    mat[1][3] = new Node(0,2);
    //the 3-d row
    mat[2][0] = new Node(4,10);
    mat[2][1] = new Node(3,1);
    mat[2][2] = new Node(1,4);
    mat[2][3] = new Node(0,8);
    //the 4-th row
    mat[3][0] = new Node(2,0);
    mat[3][1] = new Node(3,0);
    mat[3][2] = new Node(5,0);
    mat[3][3] = new Node(0,0);
    return mat;
}
```

```
public class Point{
    private int x, y;
    public Point(int x, int y){
        this.x = x;
        this.y = y;
    }
    public Point(){
        x = 0;
        y = 0;
    }
    public int x(){
        return x;
    }
    public int y(){
        return y;
    }
    public String toString(){
        return "("+x+","+y+")";
    }
}
```

בהצלחה!