

סמסטר א', מועד א', יום שני ג' שבט תשע"ז, Monday, January 30, 2017

**מחלקה:** מדעי המחשב  
**מרצה:** פרופ' ואדים (דוד) לויט, גב' אליזבת איצקוביץ  
**שם הקורס:** אלגוריתמים 1  
**מס' הקורס:** 2-7027010-1,2,3,4,5  
**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 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:

- (1) לפני תחילת העבודה עושים **Restart** למחשב.
- (2) את **workspace של java** יש לפתוח בדיסק D: שלא נמחק ב-**restart**.
- (3) **בתוך Eclipse:**  
File -> Switch Workspace -> Other -> Browse ->  
בוחרים בדיסק D:  
Where "**D:\**" workspace should be your default workspace!

- (4) יש לבנות **java project** בשם **Exam**.
- (5) אין להגדיר חבילות (**package**), יש לעבוד ב- **default package**.
- (6) לפתרון של כל שאלה צריך לבנות מחלקה נפרדת.
- (7) את קובץ הפרויקט ואת השאלון ניתן להוריד מ"מתזמן מבחנים".

**המבחן שלא יעמוד בדרשות אלו לא יזכה בנקודות!**

הפקולטה למדעי הטבע - המחלקה למדעי המחשב

בעיה מס' 1

יישמו את האלגוריתם:

**קלט:** "דף חשבוני" (עם משקלים על הצלעות) בגודל  $M$  על  $N$ ,

ונקודה  $A$  עם קואורדינטות  $(p, q)$ .

**פלט:** "כן" אם הנקודה  $A$  שייכת למסלול קצר ביותר (אחד לפחות)

בין שני הקודקודים הבאים:  $(0, 0)$  ו-  $(N, M)$ .

"לא" אם הנקודה  $A$  לא שייכת לאף מסלול קצר ביותר

בין  $(0, 0)$  לבין  $(N, M)$ .

במילים אחרות, הקלט הוא מערך דו-ממדי (מטריצה) של קודקודים. כל קודקוד (Node) מכיל את המשקלים של שתי הצלעות היוצאות ממנו לכיוונים ימינה ומעלה:



$x$  - משקל של הצלע האופקי, (הכיוון, כמו בציר ה- $X$ , ימינה)

$y$  - משקל של הצלע האנכי, (כיוון כלפי מעלה, כמו בציר ה- $Y$ )

**דוגמה:** במטריצה שלהלן:

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

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

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

	2	3	5	(3,3)
(3,0)	10 4	1 3	4 1	8
(2,0)	5 2	11 5	1 3	2
(1,0)	3 1	4 8	8 3	4
(0,0)	(0,1)	(0,2)	(0,3)	

הפקולטה למדעי הטבע - המחלקה למדעי המחשב

יש לכתוב מחלקה בשם **Q1**. יש לכתוב בנאי המחלקה ושיטה אחת.  
מבנה של קדקוד אחד מיוצג ע"י מחלקת **Node**:

```
class Node{
    int x, y, price;
    public Node(int x, int y){
        this.x = x;
        this.y = y;
        this.price = 0;
    }
}
```

מבנה של נקודה אחת מיוצג ע"י מחלקת **Point**:

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

ניתן להוסיף ל-Node כל השדות הנוספים הנדרשים ליישום האלגוריתם.

1. בנאי המחלקה מקבל מטריצה של משקלים כמערך דו-ממדי  
ונקודה **A** - אובייקט מטיפוס **Point**.

```
public Q1(Node[][] nodes, Point A){...}
```

2. שיטה שמחזירה **true** אם **A** שייכת למסלול קצר ביותר אחד לפחות בין  $(0,0)$  ו-  $(N,M)$ :

```
public boolean belongsToCheapestPath(){...}
```

אלגוריתם, סיבוכיות, דוגמא והוכחות - יש לכתוב בקובץ **WORD**.



35

**בעיה מס' 2:** נתונה סדרת מספרים שלמים  $a_1, a_2, \dots, a_{n-1}, a_n$

יש למצוא את תת-סדרה ארוכה ביותר שהפרש בין שני איברים סמוכים שלה לא יעלה על 1,

כלומר  $|a_{i-1} - a_i| \leq 1$  כאשר  $i = 2, \dots, n$ .

**דוגמה 1:**

**קלט:** 1, 2, 3, 4, 3, 2

**פלט:** 6 (כל הסדרה מקיימת את התנאי  $|a_{i-1} - a_i| \leq 1$ )

**דוגמה 2:**

**קלט:** 10, 9, 4, 5, 4, 8, 6

**פלט:** 3 (תת-סדרות המקיימות את התנאי  $|a_{i-1} - a_i| \leq 1$  הן:

10, 9, 8 או 4, 5, 4 או 4, 5, 6).

יש לכתוב מחלקה בשם **Q2**. בתוך המחלקה יש לכתוב פונקציה סטטית שמקבלת מערך של מספרים שלמים ומחזירה אורכה של תת-סדרה ארוכה ביותר שהפרש בין שני איברים סמוכים שלה לא עולה על 1.

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

אלגוריתם, סיבוכיות, דוגמא והוכחות - יש לכתוב בקובץ **WORD**.

**בעיה מס' 3:** נתונה סדרת מספרים  $a_1, a_2, \dots, a_{n-1}, a_n$ . תת-סדרה של סדרה זו

$$20 + 10 + 10 = 40$$

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, 40, 5, 3, 1

**דוגמה 3:** תת-סדרה עולה-יורדת ארוכה ביותר של סדרה 1, 2, 10, 20, 32, 40, 43 היא סדרה עצמה עם חלק יורד ריק.

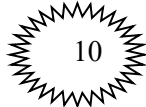
יש להכין מחלקה בשם **Q3**. בתוך המחלקה יש לכתוב את שלוש הפונקציות הסטטיות הבאות:

(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 - יש לכתוב בקובץ WORD.

**בהצלחה!**