

הפקולטה למדעי הטבע - המחלקה למדעי המחשב ומתמטיקה

סמסטר א', מועד א', יום חמישי ט"ז שבט תשע"ה, Thursday, February 5, 2015

מחלקה: מדעי המחשב ומתמטיקה

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 לדרוש 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 יזכה בנקודות!

20

בעיה מס' 1: שדה מלבני מורכב מ- $M \times N$ משבצות, חלקן צבועות בשחור. ידוע שכל המשבצות השחורות עשיות להרכיב כמה מלבנים שחורים שאינם נוגעים לא בקודקודים ולא בצלעות. יש לחשב את מספר המלבנים השחורים הנ"ל.

יש לכתוב מחלקה בשם **Question1**. בתוך המחלקה יש לכתוב פונקציה סטטית המקבלת מטריצה שמורכבת מאפסים ואחדות: 1 מתאר צבע שחור, 0 – צבע לבן. הפונקציה מחזירה את מספר המלבנים השחורים.

```
public static int numRectangles(int[][] mat){...}
```

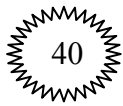
דוגמה:

קלט:

1	0	1	0	0
0	0	1	0	1
0	0	0	0	0
0	0	0	1	1
1	1	0	1	1

פלט: 5

אלגוריתם, סיבוכיות, דוגמא והוכחות.



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

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

היא 1, 2, 10, 20, 32, 40, 43

יש לכתוב מחלקה בשם **Question2**. בתוך המחלקה יש לכתוב שתי פונקציות סטטיות הבאות:

(א) פונקציה סטטית המקבלת מערך של מספרים שלמים ומחזירה אורכו של תת-מערך **עולה** ארוך ביותר בסיבוכיות $n(\log_2 n)$.

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

(ב) פונקציה סטטית המקבלת מערך של מספרים שלמים ומחזירה אורכו של תת-מערך **עולה-יורד** ארוך ביותר בסיבוכיות $n(\log_2 n)$.

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

אלגוריתם, סיבוכיות, דוגמא והוכחות.



40

בעיה מס' 3: יישמו את האלגוריתם המחשב את מספר המסלולים הקצרים ביותר בין שני קודקודים מסוימים בדף חשבוני בגודל M על N (עם משקלים מונחים על הצלעות). בנוסף, יש לחשב את מספר המסלולים האופטימאליים. המסלול נקרא אופטימאלי אם, מצד אחד, הוא קצר ביותר, ומצד שני, מספר הפניות שלו (פניה זהו שינוי הכיוון) הוא קטן ביותר. שימו לב: יש לחשב את מספר המסלולים האופטימאליים רק עם מספר המסלולים הקצרים ביותר לא עולה על מספר מסוים ($teta$) נתון מראש.

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

x - משקל של הצלע האופקי, (הכיוון, כמו בציר ה- X , ימינה)
 y - עלות 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)
	10 4	1 3	4 1	8
	5 2	11 5	1 3	2
	3 1	4 8	8 3	4
(0,0)				

בדוגמה זו עלות המסלול הזול ביותר היא 20. מספר המסלולים הקצרים ביותר הוא 4:

010110, 100110, 100101, 010101

ויש רק מסלול אופטימאלי אחד: 100110

הקוד שמבצע מילוי מטריצה זו נמצא בנספח קוד המצורף למבחן.

הפקולטה למדעי הטבע - המחלקה למדעי המחשב ומתמטיקה

יש לכתוב מחלקה בשם **Question3**. יש לכתוב בנאי המחלקה ושתי שיטות.

1. בנאי המחלקה מקבל מטריצה של משקלים כמערך דו-ממדי של קדקודים ומספר שלם **teta** (יש לשמור את **teta** כמשתנה עצם של המחלקה):

```
public Question3(Node[][] nodes, int tetat){...}
```

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 מספר המסלולים הקצרים ביותר:

```
public int numOfCheapestPaths()
```

3. שיטה שמחזירה מספר המסלולים האופטימליים כאשר מספר המסלולים הקצרים ביותר לא עולה על **teta**, אחרת היא מחזירה -1.

:

```
public int numOfOptimalPaths()
```

הפונקציה מחזירה מספר מסלולים אופטימאליים

אלגוריתם, סיבוכיות, דוגמא והוכחות

נספח קוד המצורף למבחן

```
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;
}
}
```

בהצלחה!