

## מבני נתונים - סמסטר ב' תש"פ

## מטלה 3

הנחיות:

- מטלה זו הינה להגשה ביחידים. חל איסור מוחלט למסור קוד או פתרונות לכל גורם אחר, לשום מטרה. העתקה תגרור **לפסילה מלאה** של המטלה למעתיק והמועתק.
- המטלה מחולקת לשני חלקים: חלק תיאורטי (שאלות 1,2), וחלק מעשי (שאלות 3,4). עליכם לפתור את החלק התיאורטי ולצרף אותו כקובץ PDF או תמונה. את החלק המעשי יש לצרף כשני קבצי java בשמות MyQueue.java, Ex3.java. יש להגיש את המטלה בקובץ ZIP (ולא כל דחיסה אחרת) המכיל את שלושת הקבצים הנ"ל בלבד. שם קובץ ה-ZIP יהיה מספר ת.ז. של התלמיד. אין להגיש קבצים או תיקיות מיותרים. סטייה מהנחיות אלו תגרור הורדה בציון.
- למטלה מצורפות המחלקות MyNode, MyStack. אין לשנות את המחלקות האלו (מלבד להוסיף שייוך לחבילה, במידת הצורך), ואין להגיש אותם.

שאלה 1:

נתון האלגוריתם הבא המקבל שורשים של שני עצים:

```
public boolean func(Node t1, Node t2) {
    if (t1 == null || t2 == null) return t1 == null && t2 == null;
    return func(t1.getLeft(), t2.getLeft()) && func(t1.getRight(), t2.getRight()) }
```

א. מה האלגוריתם עושה?

ב. מה הסיבוכיות שלו?

שאלה 2:

הגדרה: יהי  $T$  עץ, שקבוצת הקדקודים שלו,  $V$ , מכילה  $n$  קדקודים. העומק הממוצע של קדקוד בעץ  $T$

הוא  $\frac{1}{n} \cdot \sum_{x \in V} \text{depth}(x)$ . (כלומר ממוצע העומקים של הקדקודים בעץ). יהי  $T$  עץ בינארי בעל  $n$

קדקודים, כך שהעומק הממוצע של קדקוד ב- $T$  הינו  $\theta(\log n)$ . יהי  $h$  גובהו של  $T$ .

הוכיחו כי  $h = O(\sqrt{n \log n})$ .

רמז: התבוננו במסילה מסוימת באורך  $h$ , ותהי  $P$  קבוצת הקדקדים המשתתפים בה. תהי  $Q$  קבוצת כל יתר הקדקודים. השתמשו באי שוויון הטריויאלי הבא:

$$\frac{1}{n} \left( \sum_{x \in P} \text{depth}(x) + \sum_{y \in Q} \text{depth}(y) \right) \geq \frac{1}{n} \sum_{x \in P} \text{depth}(x)$$

### שאלה 3:

כתבו מחלקה Ex3.java ובתוכה פונקציה הבודקת האם עץ בינארי הוא שלם. (זאת אומרת, אם לכל קדקוד שאינו עלה יש שני בנים, וכל העלים באותו עומק.) אם העץ אינו שלם על הפונקציה להחזיר -2, ואם הוא שלם על הפונקציה להחזיר את גובה העץ. על הפונקציה לרוץ בסיבוכיות  $O(n)$ .

מוסכמה: גובה של עץ ריק הוא -1, וגובה של עץ המכיל רק שורש הוא 0. חובה להשתמש במחלקת צומת המצורפת. הגדרת הפונקציה:

```
public static int isFull(MyNode root)
```

### שאלה 4:

כתבו מחלקה בשם MyQueue.java הממשת תור המאחסן מספרים שלמים. לתור בנאי (constructor) המקבל כפרמטר את קיבולת התור.

```
public MyQueue(int capacity)
```

התור יאחסן את המידע באמצעות שתי מחסניות MyStack, להם פונקציות push, pop בלבד. כלומר, לא ניתן להצהיר או להשתמש במחלקה באף מבנה אחר (כגון מערך, רשימה מקושרת וכו').

```
public boolean enqueue(int data)
```

להכנסת איבר לתור בזמן ריצה  $O(1)$ , ואת הפונקציה

```
public int dequeue()
```

למחיקת והחזרת איבר בזמן ריצה  $O(n)$ .

הערות: אם התור מלא עד לקיבולת, הפונקציה enqueue תחזיר false. אם התור ריק, הפונקציה dequeue תחזיר Integer.MAX\_VALUE.

**ב ה צ ל ח ה !**