

## מבני נתונים - סמסטר קיץ תש"פ

### מטלה 1

#### הנחיות:

- מטלה זו הינה להגשה ביחידים. העתקה תגרור **לפסילה מלאה** של המטלה למעתיק והמועתק.
- המטלה מחולקת לשני חלקים: חלק תיאורטי (שאלות 1,2,3), וחלק מעשי (שאלות 4,5). עליכם לפתור את החלק התיאורטי ולצרף אותו כקובץ PDF או תמונה. את החלק המעשי יש לצרף כקובץ java בשם Range.java. יש להגיש את המטלה בקובץ ZIP (ולא כל דחיסה אחרת) המכיל את שני 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.

#### שאלה 1:

- א) ✓ סיבוכיות של אלגוריתם A היא  $O(n \log_2 n)$ . לביצוע אלגוריתם על מערך בגודל 1000 איברים דורש בדיוק 1 מילישניות (1 ms). הנניח שזמן ריצה האלגוריתם  $T(n)$  נמצא ביחס ישיר ל  $n \log_2 n$ , כלומר  $T(n) = c \cdot n \log_2 n$ .
- א. 1. עלכם לתת נוסחה לזמן ביצוע האלגוריתם  $T(k)$  עבור מערך בגודל k איברים.
- א. 2. מהו זמן ריצה של האלגוריתם עבור מערך בגודל 1000000 איברים.

- ב) ✓ האם התכונות הבאות של "Big – O" נכונות? אם כן – תוכיח את הנכונות, אם לא – תן דוגמה נגדית

1.  $O(f + g) = O(f) + O(g)$

2.  $O(f \cdot g) = O(f) \cdot O(g)$

3.  $if (g = O(f) \text{ and } f = O(h)) \text{ then } g = O(h)$

4. האם זה נכון ש-  $3\sqrt{n} + 20n + 8n^3 = O(n^4)$ ?

5. האם זה נכון ש-  $n + n^2 + n^3 = O(n^2 \log_2 n)$ ?

#### שאלה 2

- א) ✓ זמן ריצה של אלגוריתם מסוים מתואר ע"י נוסחה רקורסיבית:

$$T(n) = k \cdot T\left(\frac{n}{k}\right) + c \cdot n, \quad T(0) = 0$$

- יש למצוא את הנוסחה עבור זמן ריצה האלגוריתם במונחים של  $n, k, c$ .
- מהי הסיבוכיות של האלגוריתם במובן של "Big – O"?

(ב) מהי הסיבוכיות של הקוד הרקורסיבי הבא

```
public static func(int[]a, int low, int high, int goal){
    int t = get(a, low, high);
    if ( goal < t) return func(a, low, t-1, goal);
    else if (goal > t) return func(a, t+1, high, goal);
    else return a[t];
}
```

הטווח של המשתנים:  $0 \leq low \leq goal \leq high \leq a.length - 1$

שי להתייחס לשני מיקרים:

ב.1 סיבוכיות הפונקציה לא רקורסיבית get היא  $T(m) = cm$ , כאשר  $m = high - low + 1$

היא מחזירה מספר  $t : low \leq t \leq high$  שהוא מספר אקראי.

ב.2 סיבוכיות הפונקציה לא רקורסיבית get היא  $T(m) = 1$  והיא מחזירה מספר

$$t = (low + high)/2 : t$$

**שאלה 3 בשאלה זו יש לכתוב סיבוכיות על כל מתודה שכותבים אותה.**

א. נתחו את זמן הריצה של קטע הקוד הבא:

```
public static void Complexity(int n) {
    int x = 2;
    while( x < n ){
        for(int i=0; Math.pow(i,2) <100; i=f(i))
            System.out.println("");
        x = (int)(Math.pow(x,3));
    }
}

public static int f(int x) { return x + 2; }
```

ב. סדרו את הפונקציות לפי סדר אסימפטוטי מן "הקטנה" ל"גדולה". אם מתקיים כי  $f_i = \Theta(f_j)$  אז ציינו זאת.

הוכיחו באופן פורמלי את תשוביתכם

$f_4(n) = n!$	$f_3(n) = 2^{\log_{\sqrt{3}} n}$	$f_2 = \frac{1}{\sqrt{n}} \cdot \log_{10} n$	$f_1(n) = 2020^{2020}$
$f_8 = 8^n$	$f_7(n) = 3n^{19} + 2020n$	$f_6(n) = \log_3(3^2 n^3)$	$f_5(n) = n^n$

ג. בהינתן מערך של  $n$  מספרים בתחום  $[1, n^7]$ , מצאו אלגוריתם לינארי למיון המערך, הסבירו בקצרה (מקסימום 2 שורות לכל אלגוריתם) מה הסיבוכיות באלגוריתמי InsertionSort, BubbleSort, QuickSort, MergeSort

שאלה 4 בשאלה זו יש לכתוב סיבוכיות על כל מתודה שכותבים אותה.

בהינתן המחלקה הבאה:

```
class Person {
    public int age;
    public String name; // small letters only
    ...
}
```

א. יש לממש פונקצייה אשר בהינתן מערך של אנשים מסדרת את הילדים (מתחת לגיל 18) בתחילת במערך ואת האנשים שלפחות בגיל 18 בסוף המערך בסיבוכיות לינארית.

```
public static void arrange(Person[] persons) { . . . }
```

ב. יש לממש פונקצייה אשר בהינתן מערך של אנשים בודקת האם המערך הוא סימטרי ביחס לגיל של האנשים, ממשו בעזרת מחסנית עזר (אפשר להשתמש בספרייה utils) לדוגמא:

קלט: {12,4,81,3,3,81,4,12} כאשר המספר  $i$  מייצג אובייקט Person עם  $age=i$

פלט: true

קלט: {12,2} כאשר המספר  $i$  מייצג אובייקט Person עם  $age=i$

פלט: false

```
public static boolean isSymmetrical(Person[] persons){...}
```

ג. ענו על סעיף ב' בעזרת תור ובעזרת מחסנית עזר (אפשר להשתמש בספרייה utils)

ד. רשמו פסאדו קוד או הסבירו במילים כיצד לכתוב פונקצייה אשר ממיינת מערך של אנשים לפי

שם פרטי בעזרת מיון Radix Sort

שאלה 5 בשאלה זו יש לכתוב סיבוכיות על כל מתודה שכותבים אותה.

א. יש לממש מחלקה גנרית בשם MyQueue המתארת תור בעזרת רשימה מקושרת דו-כיוונית.

המחלקה מכילה מתודות הבאות:

- enqueue(T val) להחסנת איבר בתור,
- T dequeue() לשליפת איבר מתור,
- boolean contains(t val) המתודה מחזירה true אם האיבר נמצא בתור, אחרת היא מחזירה false. (בעזרת equals)
- boolean isEmpty() המתודה מחזירה true אם התור הוא ריק, אחרת היא מחזירה false.
- int size() המתודה מחזירה מספר איברים שנמצאים בתור.

- ב. יש לממש מחלקה **גנרית** בשם `MyStack` המתארת מחסנית בעזרת רשימה מקושרת דו-כיוונית. המחלקה מכילה מתודות הבאות:
- push(T val) להחסנת איבר במחסנית,
  - T pop() לשליפת איבר מתור,
  - boolean contains(t val) המתודה מחזירה true אם האיבר נמצא במחסנית, אחרת היא מחזירה false. (בעזרת equals)
  - boolean isEmpty() המתודה מחזירה true אם המחסנית היא ריקה, אחרת היא מחזירה false.
  - int size() המתודה מחזירה מספר איברים שנמצאים במחסנית.

- ג. כמה מחסניות אנו צריכים כדי לממש תור? נמק היטב.  
יש לממש תור בעזרת אותו מספר של מחסניות. אסור להשתמש מבנים אחרים, פרט ממחסנית.

- ד. כמה תורים אנו צריכים כדי לממש מחסנית?  
יש לממש מחסנית בעזרת אותו מספר של תורים. אסור להשתמש מבנים אחרים, פרט מתור

**בהצלחה רבה!**

תשובה א.

$$T(n) = cn \log_2 n \rightarrow T(k) = ck \log_2 k \rightarrow c = \frac{T(k)}{k \log_2 k}$$

$$\rightarrow T(n) = \frac{T(k)n \log_2 n}{k \log_2 k} = \frac{T(k)n \log_k n}{k}$$

לכן

$$T(1000000) = \frac{T(1000) \cdot 1000000 \cdot \log_{1000} 1000000}{1000} = 2000ms = 2sc$$

שאלה 2:

שאלה 3:

בהצלחה!