

מבני נתונים – סמסטר ב' תש"פ
מטלה 1

הנחיות:

- מטלה זו הינה ביחידים, העתקה תגרור **לפסילה מלאה** של המטלה למעתיק והמועתק ממנו.
- המטלה מחולקת לשני חלקים: חלק תיאורטי (שאלות 1-3, ושאלה 4 סעיף ב'), וחלק מעשי (שאלה 4 סעיף א'), עליכם לפתור את החלק התיאורטי ולצרף אותו כקובץ PDF או תמונה, ואת החלק המעשי לצרף אותו כקובץ java בשם Ex1.java, את שני החלקים יש לדחוס כקובץ ZIP (ולא שום דחיסה אחרת) בשם ת"ז של הסטודנט. סטייה מהדרכות אלו תגרור הורדה בציון.
- בשאלות שנדרשים בהם לחשב זמן ריצה של פונקציה מסוימת, הכוונה הינה לסדר גודל של זמן הריצה - $O(\dots)$, כפי שלמדתם בהרצאה.

שאלה 1: לכל אחת מן הטענות הבאות, קבעו אם היא נכונה או לא נכונה, והוכיחו את תשובתכם.

- **טענה 1:** יהיו $f(n) = (\sqrt{n})^{\sqrt{n}}$, $g(n) = \lfloor \sqrt{n} \rfloor!$ אזי:

$$\log(f(n)) = \Theta(\log(g(n)))$$

כלומר:

$$\log(f(n)) = O(\log(g(n)))$$

וגם

$$\log(f(n)) = \Omega(\log(g(n)))$$

- **טענה 2:** תהי T פונקציה המוגדרת רקורסיבית באופן הבא:

$$T(n) = \left(T\left(\frac{n}{2}\right)\right)^2 \cdot 2^n, T(1) = 2$$

אזי:

$$T(n) = 2^{\Theta(n)}$$

כלומר:

$$T(n) = 2^{O(n)}$$

וגם

$$T(n) = 2^{\Omega(n)}$$

- **טענה 3:** אם הפונקציות f ו-g הן פונקציות מונוטוניות עולות כך שמתקיים:

$$f(g(n)) = O(n), \quad f(n) = \Omega(n)$$

אזי:

$$g(n) = O(n)$$

שאלה 2:

נתונות הפונקציות הבאות:

$f_1(n) = 2019^{2020}$	$f_2(n) = 2^{3^n}$	$f_3(n) = \frac{1}{n^{2020}}$
$f_4(n) = 2^{\log_{\sqrt{2}} n}$	$f_5(n) = 4^n$	$f_6(n) = n^n$
$f_7(n) = \log_3(3^n n^2)$		

סדרו את הפונקציות לפי סדר אסימפטוטי $O(\dots)$, מן ה"קטנה" ל"גדולה". אם מתקיים $f_i = \Theta(f_j)$ ציינו זאת, הוכיחו פורמלית את תשובותיכם.

שאלה 3: חשבו את זמני הריצה של הפונקציות הבאות:

```
//A:
public static void mystery1(int n) {
    int k = n;
    while (k > 2) {
        System.out.println(k);
        k = (int) (Math.pow(k, 1 / 3));
    }
}
```

```
// B:
public static void dunno1(int n) {
    int i, j;
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= i; j++) {
            mystery1(n);
        }
    }
}
```

```
//C:
public static void mystery2(int i) {
    int x;
    x = 1;
    while (x < i) {
        System.out.println(x);
        x = x * 2;
    }
    while (x > 2) {
```

<pre> x = (int) (Math.pow(x, 1 / 2)); System.out.println(x); } } </pre>
<pre> // D: public static void dunno2(int n) { int i; for (i = n; i >= 1; i--) { mystery2(i); } } </pre>

- הסבירו את תשובתכם בצורה פורמלית ומפורטת ככל האפשר.

שאלה 4:

- א. כתבו אלגוריתם (וממשו בשפת JAVA) שמקבל מערך ממיון ומספר k , ומחזיר את האינדקס של k במערך, אם k לא נמצא במערך, יש להחזיר -1.
- ב. נתחו את זמן הריצה של הפונקציה שכתבתם.

חותמת הפונקציה:

```
public static int Search(int [] arr, int k)
```

דרישות:

- זמן הריצה צריך להיות $O(\log(m))$, כאשר m הוא מספר האיברים שמופיעים לפני k במערך הנתון - אם נמצא, אם לא, m הוא מספר האיברים שמופיעים לפני המספר הכי קרוב ל- k במערך הנתון.

רמז: תחשבו על החיפוש בצעדים שהולכים וגדילים בקצב אקספוננציאלי, לאו דווקא מאמצע המערך.

בהצלחה !