# מבני נתונים - תרגיל 1

**תאריך פרסום** : 29.03.2022

**תאריך הגשה** : 23:59 ,12.04.2022

**מרצה ומתרגל אחראי** : מיכל שמש, יותם אשכנזי

## הנחיות:

- הגשת העבודה הינה ביחידים או בזוגות, לבחירתכם. אנו ממליצים לעבוד בזוגות, כדי לעודד דיון והפריה הדדית. למען הסר ספק, אם העבודה הוגשה בזוג אז כל סעיפי העבודה המוגשים צריכים להיות תוצאה של עבודה משותפת.
- חובה להגדיר קבוצה באתר המודל ולהשתייך אליה על מנת להגיש את עבודה 1, גם אם בחרתם להגיש לבד (ואז אתם בקבוצה המכילה רק אתכם). למידע נוסף בקרו בעמוד ההרשמה לקבוצה.
- העבודה חייבת להיות מוקלדת ומוגשת כקובץ בפורמט Pdf. שם הקובץ שלכם צריך להיות זהה לשם הקבוצה אליה נרשמתם באתר המודל, לדוגמא : Assignment1_Group300.pdf.
- שאלות לגבי העבודה יש לשאול בפורום הייעודי באתר המודל או בשעות קבלה של המרצה\המתרגל האחראי על העבודה. הפורום נועד לדיון בין הסטודנטים בנוגע לעבודה, הצוות האחראי על העבודה יענה על שאלות בפורום רק במידה שהן דורשות הבהרה. אחת ל-24 שעות צוות העבודה יקרא את השאלות שעלו בפורום וככל שיידרשו יפרסם הבהרות בשרשור ייעודי.
- אין צורך לפרט דברים שנלמדו כיתה. עם זאת, יש להוכיח כל טענה שלא נלמדה בהרצאה או בתרגול.
- יש לנתח את זמן הריצה בצורה ההדוקה ביותר תוך התחשבות במקרה הגרוע ביותר.
- log() מתייחס ללוגריתם בבסיס 2.
- יש לתת הוכחות פורמליות עבור חסמים אסימפטוטיים, בדומה לדוגמאות שהועברו בכיתה.

## שאלה 1 – היררכיית סדרי גודל

נתונות לכם הפונקציות הבאות ועליכם למיין אותן בטבלה הנתונה לפי סדר אסימפטוטי מן ה"קטנה" ל"גדולה" כך ש:

- כל זוג פונקציות $f_i, f_k$ באותה השורה יקיימו $f_i(n) = \Theta(f_k(n))$
- לכל שתי פונקציות בשורות עוקבות יתקיים $f_i(n) = O(f_k(n))$ (עבור $f_i(n)$ בשורה ה-$j$ ו-$f_k(n)$ בשורה ה-$j+1$).

עליכם להוכיח פורמאלית את תשובותיכם באופן הבא:

- בשלב הראשון עליכם לציין את סדר הגודל של כל פונקציה (קרי, מיהי $g(n)$, עבורה מתקיים $f_i(n) = \Theta(g(n))$), ולהוכיח זאת לפי ההגדרה.
- בשלב השני עליכם לציין אילו פונקציות מקיימות $f_i(n) = \Theta(f_k(n))$, ולהוכיח זאת לפי ההגדרה.
- בשלב השלישי, עליכם להוכיח לפי ההגדרה את הסדר של הפונקציות בטבלה (כלומר שכל שורה מהווה חסם עליון לשורה שלפניה).

<u>הפונקציות:</u>

$$f_1(n) = 2022, \quad f_2(n) = \log(n^8), \quad f_3(n) = 2^{\sqrt{n}}, \quad f_4(n) = 3n^3 + 2\log(n) + 1,$$

$$f_5(n) = 4^{(2^n)}, \quad f_6(n) = \frac{1}{\sqrt{n}}, \quad f_7(n) = 6^{\log_{\sqrt{6}} n}, \quad f_8(n) = 2^{64}, \quad f_9(n) = 2^{(4^n)},$$

$$f_{10}(n) = n^n, \quad f_{11}(n) = \log_2(2^n n^2), \quad f_{12}(n) = \log\left(n^{\frac{1}{2}}\right), \quad f_{13}(n) = 4^n,$$

$$f_{14}(n) = \frac{2n}{7}, \quad f_{15}(n) = \frac{1}{3n^2}$$

<u>טבלת מיון לפי סדרי גודל:</u>

| סדר גודל אסימפטוטי (מהקטן לגדול) | הפונקציות |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

## שאלה 2 – תכונות של חסמים אסימפטוטיים

יהיו $f(n), g(n)$ פונקציות חיוביות. בסעיפים א׳-ד׳ הוכיחו שהפונקציות מקיימות את התכונות הבאות:

א. טרנזיטיביות (Transitivity):

אם $f(n) = \Theta(g(n))$ וגם $g(n) = \Theta(h(n))$, אז $f(n) = \Theta(h(n))$.

ב. רפלקסיביות (Reflexivity):

a. $f(n) = \Theta(f(n))$

b. $f(n) = \Omega(f(n))$

c. $f(n) = O(f(n))$

ג. סימטריות (Symmetry):

$f(n) = \Theta(g(n))$ אמ״ם $g(n) = \Theta(f(n))$.

ד. אנטי-סימטריות (Transpose symmetry):

$f(n) = O(g(n))$ אמ״ם $g(n) = \Omega(f(n))$.

ה. יהיו $p_1(n), p_2(n)$ פולינומים מחזקה $n_1, n_2$ בהתאמה. מיצאו פונקציה חיובית $g(n)$ כך שמתקיים $p_1(p_2(n)) = \Theta(g(n))$. הוכיחו את תשובתכם.

## שאלה 3 – פתרון נוסחאות נסיגה

מצאו חסם אסימפטוטי הדוק $\Theta$ עבור $T(n)$ בכל אחת מנוסחאות הנסיגה שלהלן. הניחו כי $T(n)$ קבועה עבור $n$ קבוע. הוכיחו את תשובותיכם. ניתן להשתמש בכל אחת מהשיטות שנלמדו בכתה.

א. $T(n) = T(\sqrt{n}) + 1$

ב. $T(n) = 2T\left(\frac{n}{2}\right) + n$

ג. $T(n) = 4T\left(\frac{n}{2}\right) + n^3 \log n$

ד. $T(n) = T\left(\frac{2}{5}n\right) + 1$

ה. $T(n) = T(cn) + T((1-c)n) + 1$, $0 < c < 1$.

ו. $T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n$

ז. $T(n) = 6T\left(\frac{n}{3}\right) + n$

## שאלה 4 – ניתוח זמן ריצה

מהי סיבוכיות זמן הריצה $T(n)$ של קטעי הקוד הבאים (במונחים של $\Theta$)? נסחו את תשובותיכם באופן מפורש (דהיינו ללא שימוש בסכומים, מכפלות או סימן עצרת). הסבירו ונמקו* את דרך ההגעה לפתרון.

*יש לנתח את זמן הריצה כפי שהודגם בכתה עבור הניתוח של אלגוריתם מיון-הכנסה. עבור אלגוריתמים רקורסיביים יש לנסח נוסחת נסיגה שמתארת את זמן הריצה ולפתור אותה באחת מהשיטות שנלמדו בכתה.

a) Input: An array of numbers A, |A|=n and a number key.

```
function Foo1(A, key)
    index ← -1
    for i←0 to n-1
        if A[i] == key
            index←i
            break
    return index
```

b) Input: An array of numbers A, |A|=n.

```
function Foo2 (A)
    for i ← 0 to n-1
        for j ← n-1 to 0
            if (A[i] > A[j])
                A[i] = j
            else
                A[j] = i
```

c) Input: 2 numbers – n and base.

```
function exp(base , n)
    if (n = 0)
        return 1
    else if (n = 1)
        return base
    else
        return base · exp(base, n-1)
```

d)

**רמז**: בשלב הראשון ניתן למצוא נוסחה נפרדת לכל אחד מהמקרים השונים (מקרים עבור ערכי קלט). לאחר מכן ניתן למצוא נוסחה אחת שתתאים לכל n ולהמשיך כרגיל בהוכחה כפי שלמדנו.

Input: 2 numbers – n and base.

```
function exp2(base , n)
    if (n = 0)
        return 1
    else if (n = 1)
        return base
    else if (mod(n, 3) = 0)
        tmp ← exp2(base, n/3)
        return tmp · tmp · tmp
    else
        return base · exp2(base,n-1)
```

e)

נתון מספר טבעי $n \geq c \geq 2$. בסעיף זה יש לנתח את זמן הריצה כתלות ב-$c$.
מהו זמן הריצה כאשר $c$ שווה ל-$n$ ? מהו זמן הריצה כאשר $c$ קבוע?

**רמז** : בדומה לסעיף הקודם, ניתן להתחיל עם c-1 נוסחאות התחלתיות. רמז נוסף : חישבו מהו היחס בין שני המקרים המוצגים בסעיפים d,e וכיצד ניתן להשתמש בו.

Input: 2 numbers – n and base.

```
function expC(base , n)
    if (n = 0)
        return 1
    else if (n = 1)
        return base
    else if (mod(n, c) = 0)
        tmp ← expC(base, n/c)
        ans ← 1
        for i ← 1 to c:
            ans ← ans * tmp
        return ans
    else
        return  base · expC(base,n-1)
```

## שאלה 5 – פיתוח אלגוריתמים

א. <u>נתונה הבעיה הבאה</u>:

יהי $N>1$ מספר טבעי כלשהו.

<u>קלט</u>: 2 מערכים A,B לא ממוינים באורך N. כל מערך מכיל N מספרים מ 0 עד 2N.

<u>פלט</u>: האלגוריתם יחזיר TRUE אם כל המספרים במערך A שונים מכל המספרים במערך B (כלומר אין אף איבר משותף לשניהם). שימו לב שיכולים להיות במערך מסוים שני תאים עם אותו מספר ועדין לקבל TRUE במידה שבמערך השני המספרים שונים.

תארו אלגוריתם (בפסאודו-קוד) בעל זמן ריצה יעיל ככל הניתן לבעיה. הביעו את זמן הריצה של האלגוריתם (במונחים של Θ) והוכיחו את תשובתכם. אין צורך להתייחס למקום בזיכרון הנדרש עבור מימוש האלגוריתם.

לדוגמה, בהינתן המערכים A ו B בגודל 7

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A: | 5 | 2 | 14 | 2 | 2 | 7 | 1 |
| B: | 6 | 9 | 9 | 10 | 13 | 0 | 8 |

על האלגוריתם להחזיר: TRUE (שימו לב שהמספר 2 מופיע יותר מפעם אחת במערך A)

דוגמה נוספת, בהינתן המערכים A ו B בגודל 7

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A: | 5 | 2 | 14 | 2 | 2 | 7 | 1 |
| B: | 6 | 9 | 9 | 10 | 7 | 13 | 8 |

על האלגוריתם להחזיר: FALSE משום שהאיבר 7 מופיע בשני המערכים

ב. <u>נתונה הבעיה הבאה</u>:

<u>קלט</u>: מערך ממוין A באורך N ומספר כלשהו x.

<u>פלט</u>: האינדקס של x במערך הממויין A. אם x לא קיים במערך יש להחזיר -1.

תארו אלגוריתם (בפסאודו-קוד) בעל זמן ריצה $O(\log d)$ כאשר d הוא מספר האיברים מופיעים לפני איבר x במערך המבוקש במידה ו x- קיים במערך, ו d שווה ל n במידה ו x אינו קיים במערך.

רמז: חשבו על החיפוש בצעדים הולכים ומשתנים בקצב מעריכי, לאו דווקא מאמצע המערך.

לדוגמה, בהינתן מערך (A,7) על המערך להחזיר 5

| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| A: | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 12 | 15 |