<u>עבודה 1 מבני נתונים</u>

<u>שאלה 1 :</u> נוכיח בעזרת גבולות את הסדר האסימפטוטי של הפונקציות הנתונות :

$$f(n) = \boldsymbol{O}\big(g(n)\big)\ if \quad \lim_{n\to\infty}\frac{f(n)}{g(n)} = 0$$

$$f(n) = \boldsymbol{\Theta}\big(g(n)\big)\ if \quad \lim_{n\to\infty}\frac{f(n)}{g(n)} = c \ \ (0 < c < \infty)$$

היחס O הוא יחס טרנזיטיבי ולכן נסתפק להוכיח את היחס אך ורק בין כל שתי פונקציות עוקבות ביחס :

$$f4(n), f1(n), f9(n)\ \Theta\ f11(n), f10(n), f12(n)\ \Theta\ f2(n), f3(n), f5(n), f7(n), f8(n), f6(n)$$

$\lim_{n\to\infty}\frac{f4(n)}{f1(n)} = \lim_{n\to\infty}\frac{\frac{1}{n^2}}{2019} = \lim_{n\to\infty}\frac{1}{2019n^2} = 0$
$\lim_{n\to\infty}\frac{f1(n)}{f9(n)} = \lim_{n\to\infty}\frac{2019}{\log(n^{\frac{2}{3}})} = 0$
$\lim_{n\to\infty}\frac{f9(n)}{f11(n)} = \lim_{n\to\infty}\frac{\log(n^{\frac{2}{3}})}{\log(n^{100})} = \lim_{n\to\infty}\frac{\frac{2}{3}\log n}{100\log n} = \frac{1}{150}$
$\lim_{n\to\infty}\frac{f11(n)}{f10(n)} = \lim_{n\to\infty}\frac{\log(n^{100})}{\log_3(3^n * n^2)} = \lim_{n\to\infty}\frac{100\log n}{n\log_3 3+2\log_3 n} = \lim_{n\to\infty}\frac{100\log n}{n+2\log_3 n} = loop\left[\frac{\infty}{\infty}\right] =$ $\lim_{n\to\infty}\frac{\frac{100}{nln2}}{1+\frac{2}{nln3}} = \lim_{n\to\infty}\frac{100ln3}{nln2ln3+2ln2} = 0$
$\lim_{n\to\infty}\frac{f10(n)}{f12(n)} = \lim_{n\to\infty}\frac{\log_3(3^n * n^2)}{n^2+n*\ln(n^{10})+n+10} = \lim_{n\to\infty}\frac{n+2\log_3 n}{n^2+10\mathrm{n}*\ln(n)+n+10} = loop\left[\frac{\infty}{\infty}\right] =$ $\lim_{n\to\infty}\frac{\frac{nln3+2}{nln3}}{2n+10\ln(n)+11} = \frac{1}{\infty} = 0$
$\lim_{n\to\infty}\frac{f12(n)}{f2(n)} = \lim_{n\to\infty}\frac{n^2+n*ln\big(n^{10}\big)+n+10}{2^{\log_{\sqrt{2}} n}} = \lim_{n\to\infty}\frac{n^2+n*ln\big(n^{10}\big)+n+10}{\big(\sqrt{2}\big)^{\log_{\sqrt{2}} n} * \big(\sqrt{2}\big)^{\log_{\sqrt{2}} n}} =$ $\lim_{n\to\infty}\frac{n^2+n*ln\big(n^{10}\big)+n+10}{n^2} = 1$
$\lim_{n\to\infty}\frac{f2(n)}{f3(n)} = \lim_{n\to\infty}\frac{n^2}{2^{\sqrt{n}}} = loop\left[\frac{\infty}{\infty}\right] = \frac{4n^{\frac{3}{2}}}{2^{\sqrt{n}}ln2} = loop\left[\frac{\infty}{\infty}\right] = \lim_{n\to\infty}\frac{12n}{2^{\sqrt{n}}\ln^2 2} = loop\left[\frac{\infty}{\infty}\right] =$ $\lim_{n\to\infty}\frac{24\sqrt{n}}{2^{\sqrt{n}}\ln^3 2} = loop\left[\frac{\infty}{\infty}\right] = \lim_{n\to\infty}\frac{24}{2^{\sqrt{n}}\ln^4 2} = 0$
$\lim_{n\to\infty}\frac{f3(n)}{f5(n)} = \lim_{n\to\infty}\frac{2^{\sqrt{n}}}{4^n} = \lim_{n\to\infty}\frac{1}{2^{2n-\sqrt{n}}} = 0$
$\lim_{n\to\infty}\frac{f5(n)}{f7(n)} = \lim_{n\to\infty}\frac{4^n}{n^n} = \lim_{n\to\infty}(\frac{4}{n})^n = \lim_{n\to\infty} e^{n*\ln(\frac{4}{n})} = e^{\lim_{n\to\infty} n*\ln(\frac{4}{n})} = 0$
$\lim_{n\to\infty}\frac{f7(n)}{f8(n)} = \lim_{n\to\infty}\frac{n^n}{3^{2^n}} = \lim_{n\to\infty}\frac{e^{nln(n)}}{e^{2^n\ln(3)}} = \lim_{n\to\infty}\frac{nln(n)}{2^n\ln(3)} = loop\left[\frac{\infty}{\infty}\right] = \lim_{n\to\infty}\frac{\ln(n)+1}{2^n\ln(2)\ln(3)} =$ $\lim_{n\to\infty}\frac{1}{n2^n\ln^2(2)\ln(3)} = 0$
$\lim_{n\to\infty}\frac{f8(n)}{f6(n)} = \lim_{n\to\infty}\frac{3^{2^n}}{2^{3^n}} = \lim_{n\to\infty}\frac{e^{2^n\ln(3)}}{e^{3^n\ln(2)}} = \frac{\ln(3)}{\ln(2)}\lim_{n\to\infty}\frac{2^n}{3^n} = 0$

<u>שאלה 2</u>: א. <u>הוכחה:</u> $f(n) = 2^{n^2}$

$$\lim_{n\to\infty} \frac{f(n-k)}{f(n)} = \lim_{n\to\infty} \frac{2^{(n-k)^2}}{2^{n^2}} = \lim_{n\to\infty} \frac{1}{2^{2nk-k^2}} = 0$$

ב. <u>נניח בשלילה</u> כי קיימת פונקציה המקיימת את התנאים הנ"ל ולכן קיימים c1,n1>0 ומתקיים לכל n>n1:

$$f^2(n) \le c1 * f(n) \to f(n) \le c1$$

בנוסף אנו יודעים כי קיימים *c2,n2>0* ומתקיים לכל *n>n2*:

$$f(n) \ge c2 * \log\log n$$

ובסה"כ קיבלנו כי עבור *n>max(n1,n2)* מתקיים:

$$c1 \ge c2 * \log\log n$$

סתירה לעצם כך ש c1 קבוע ו $c2 * \log\log n$ שואף לאינסוף.

ג. <u>הוכחה:</u> *c=2, n0=1*

$$g(n) \ge 1 \to f(n) * g(n) \ge f(n)$$

$$f(n) \ge 1 \to f(n) * g(n) \ge g(n)$$

נחבר בין שני אי השוויונות וקיבלנו כי לכל $n \ge 1$:

$$2 * f(n) * g(n) \ge f(n) + g(n)$$

ובסה"כ הראינו כי $f(n) + g(n) = \boldsymbol{o}(f(n) * g(n))$ מ.ש.ל

ד. <u>דוגמא נגדית:</u> $g(n) = n, f(n) = \frac{1}{n}$

$$\lim_{n\to\infty} \frac{f^{g(n)}(n)}{f(g(n))} = \lim_{n\to\infty} \frac{\left(\frac{1}{n}\right)^n}{\frac{1}{n}} = \lim_{n\to\infty} \frac{1}{n^{n-1}} = 0$$

שאלה 3:

א. נמצא חסם עליון ותחתון בעזרת שיטת האיטרציה:

$$T(2)=c=\mathbf{\Theta}(1)$$

$$T(n)=T\left(n^{\frac{2}{3}}\right)+17=\left[T\left(n^{\left(\frac{2}{3}\right)^2}\right)+17\right]+17=\left[T\left(n^{\left(\frac{2}{3}\right)^3}\right)+17\right]+17*2=\cdots=$$
$$\left[T\left(n^{\left(\frac{2}{3}\right)^i}\right)+17i\right]$$

$$n^{\left(\frac{2}{3}\right)^i}=2\rightarrow\left(\frac{2}{3}\right)^i*logn=1\rightarrow logn=\left(\frac{3}{2}\right)^i\rightarrow i=\log_{\frac{3}{2}}\log n$$

לאחר $i=\log_{\frac{3}{2}}\log n$ צעדים הגענו למקרה בסיס ולכן:

$$\mathrm{T(n)}=T\left(n^{\left(\frac{2}{3}\right)^i}\right)+17i=T(2)+17\log_{\frac{3}{2}}\log n=\mathbf{\Theta}(1)+\mathbf{\Theta}(\log_{\frac{3}{2}}\log n)=\mathbf{\Theta}(\log_{\frac{3}{2}}\log n)$$

ב. נמצא חסם עליון ותחתון בעזרת שיטת המאסטר (מקרה ג'):

$$a=7\,,b=2\,,f(n)=n^4*\log\log n\,,\varepsilon=1,c=\frac{7}{64}$$

$\lim\limits_{n\to\infty}\frac{f(n)}{n^{\log_b(a)+\varepsilon}}=\lim\limits_{n\to\infty}\frac{n^4*\log\log n}{n^{\log(7)+\varepsilon}}=\lim\limits_{n\to\infty}n^{0.2}*\log\log n=\infty\rightarrow f(n)=\Omega(n^{\log_b(a)+\varepsilon})$
$af\left(\frac{n}{b}\right)=7*\left(\frac{n}{2}\right)^4*\log\log\frac{n}{2}=\frac{7}{64}*n^4*\log\log\frac{n}{2}\le\frac{7}{64}*n^4*\log\log n=c*f(n)$

ובסה"כ קיבלנו כי: $T(n)=\mathbf{\Theta}\big(\mathrm{f(n)}\big)=\mathbf{\Theta}(\mathrm{n}^4*\log\log n)$

ג. נראה בעזרת אינדוקציה כי $\mathrm{T}(n)=\mathbf{\Theta}(\mathrm{n})$:

ראשית נראה כי $T'(n)=T'(cn)+T'\big((1-c)n\big)=\mathbf{\Theta}(\mathrm{n})$

<u>מקרה בסיס:</u> $T'(1)=k=\mathbf{\Theta}(1)$

<u>הנחת האינדוקציה:</u> עבור כל *k<n* קיימים *c1,c2* כך שמתקיים $c1k\le T'(k)\le c2k$

<u>צעד האינדוקציה:</u> נרצה להראות כי קיימים *d1,d2* כך ש: $d1n\le T'(n)\le d2n$

ע"פ הנחת האינדוקציה נקבל כי:

$$c1cn+c1(1-c)n\le T'(n)=T'(cn)+T'\big((1-c)n\big)\le c2cn+c2(1-c)n$$

ולכן קיימים d1=c1,d2=c2 כך ש:

$$d1n \le T'(n) \le d2n$$

ולכן: $T'(n) = \mathbf{\Theta}(\mathrm{n})$, ולסיכום: $T(n) = T'(n) + 1 = \mathbf{\Theta}(\mathrm{n}) + \mathbf{\Theta}(1) = \mathbf{\Theta}(\mathrm{n})$

ד. נראה בעזרת אינדוקציה כי $\mathrm{T}(n) = \mathbf{\Theta}(\mathrm{n})$:

<u>מקרה בסיס:</u> $T(1) = k = \mathbf{\Theta}(1)$

<u>הנחת האינדוקציה:</u> עבור כל *k<n* קיימים *c1,c2* כך שמתקיים $c1k \le T(k) \le c2k$

<u>צעד האינדוקציה:</u> נרצה להראות כי קיימים *d1,d2* כך ש: $d1n \le T(n) \le d2n$

ע"פ הנחת האינדוקציה נקבל כי:

$$c1 * \frac{2}{5}n + 3c1 * \frac{1}{5}n + n \le T(n) = T\left(\frac{2}{5}n\right) + 3T\left(\frac{1}{5}n\right) + n \le c2 * \frac{2}{5}n + 3c2 * \frac{1}{5}n + n$$

ומכאן ש:

$$(c1 + 1)n \le T(n) \le (c2 + 1)n$$

ולכן קיימים d1=c1+1,d2=c2+2 כך ש:

$$d1n \le T(n) \le d2n$$

ולסיכום: $T(n) = \mathbf{\Theta}(\mathrm{n})$

ה. נמצא חסם עליון ותחתון בעזרת שיטת האיטרציה עבור $\mathrm{T}(1) = 1 = \mathbf{\Theta}(1)$:

$$\mathrm{T(n)} = \frac{3}{2}T(n-1) + 1 = \frac{3}{2}\left[\frac{3}{2}T(n-2) + 1\right] + 1 = \left(\frac{3}{2}\right)^2 T(n-2) + \left(1 + \frac{3}{2}\right) = \left(\frac{3}{2}\right)^2\left[\frac{3}{2}T(n-3) + 1\right] + \left(1 + \frac{3}{2}\right) = \left(\frac{3}{2}\right)^3 T(n-3) + \left(1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2\right) = \cdots = \left(\frac{3}{2}\right)^i T(n-i) + \frac{1*\left(\left(\frac{3}{2}\right)^i - 1\right)}{\frac{3}{2} - 1} = \left(\frac{3}{2}\right)^i T(n-i) + 2 * \left(\frac{3}{2}\right)^i - 2$$

$$n - i = 1 \rightarrow i = n - 1$$

לאחר $i = n - 1$ צעדים הגענו למקרה בסיס ולכן:

$$\mathrm{T(n)} = \left(\frac{3}{2}\right)^i T(n-i) + 2 * \left(\frac{3}{2}\right)^2 - 2 = \left(\frac{3}{2}\right)^{n-1} T(1) + 2 * \left(\frac{3}{2}\right)^{n-1} - 2 = 3 * \left(\frac{3}{2}\right)^{n-1} - 2 = \mathbf{\Theta}\left(\left(\frac{3}{2}\right)^{\mathrm{n-1}}\right)$$

שאלה 4

א.

```
for i← 1 to n -1
    for j← n downto i+1
        if A[j-1] > A[j]
            temp ←A[j-1]
            A[j-1] ← A[j]
            A[j] ← temp
```

Times	Cost	Line
n	C1	1
$\sum_{i=1}^{n-1} n-(i+1)+1$	C2	2
$\sum_{i=1}^{n-1} n-(i+1)$	C3	3
לכל היותר: $\sum_{i=1}^{n-1} n-(i+1)$	C4	4
לכל היותר: $\sum_{i=1}^{n-1} n-(i+1)$	C5	5
לכל היותר: $\sum_{i=1}^{n-1} n-(i+1)$	C6	6

$$T(n) = n * c1 + (\textstyle\sum_{i=1}^{n-1} n-(i+1)+1) * c2 + (\sum_{i=1}^{n-1} n-(i+1)) * (c3+c4+c5+c6) = n * c1 + \frac{n(n-1)}{2} * c2 + \left(\frac{(n-1)(n-2)}{2}\right) * (c3+c4+c5+c6) = \left(\frac{c2+c3+c4+c5+c6}{2}\right) n^2 + (c1 - \frac{1}{2}c2 - \frac{3}{2}(c3+c4+c5+c6)n + (c3+c4+c5+c6) = an^2 + bn + c = \boldsymbol{\Theta}(\mathrm{n}^2)$$

ב. נסמן power=n ונשים לב כי: $T(1) = \boldsymbol{\Theta}(1)$. נגדיר את הנוסחה הרקורסיבית עבור *T(n)*:

$$T(n) = T(n-1) + c = [T(n-2) + c] + c = T(n-2) + 2c = [T(n-3) + c)] + 2c = T(n-3) + 3c = \cdots = T(n-i) + ic$$

לאחר i=n-1 צעדים הגענו למקרה בסיס ומכאן ש:

$$\mathrm{T}(n) = T(n-i) + ic = T(1) + (n-1)c = 1 + nc - c = cn + (1-c) = \boldsymbol{\Theta}(\mathrm{n})$$

```
function exp2(base , power)
    if (power = 0)
        return 1
    else if (power = 1)
        return base
    else if (mod(power, 2) = 0)
        tmp ← exp2(base, power/2)
        return tmp · tmp
    else
        return base · exp2(base,power-1)
```

ג. נסמן power=n ונשים לב כי: $T(1) = \mathbf{\Theta}(1)$.

8נשים לב כי במקרה "הטוב ביותר" (power הוא חזקה של 2) בכל קריאה של הפונקציה הרקורסיבית נקרא ל **exp2**(base, power/2), ולכן הנוסחה הרקורסיבית המתאימה היא:

$$T(n) = T\left(\frac{n}{2}\right) + \mathrm{c} = \left[T\left(\frac{n}{4}\right) + \mathrm{c}\right] + \mathrm{c} = \mathrm{T}\left(\frac{\mathrm{n}}{4}\right) + 2\mathrm{c} = \left[\mathrm{T}\left(\frac{\mathrm{n}}{8}\right) + \mathrm{c}\right] + 2\mathrm{c} = \mathrm{T}\left(\frac{\mathrm{n}}{8}\right) + 3\mathrm{c} = \cdots = T(1) + c * log(n) = \mathbf{\Theta}(\log(\mathrm{n}))$$

לעומת זאת, במקרה "הגרוע ביותר" נקרא ל **exp2**(base, power/2) ול **exp2**(base, power-1) לסירוגין (למשל עבור power=30), וזה שקול למספר הפעמים אם היינו קוראים בכל פעם ל **exp2**(base, (power-1)/2) כפול 2 (אך מכפלה בקבוע אינה משנה את הסדר האסימפטוטי של הפונקציה). מכאן, הנוסחה הרקורסיבית המתאימה היא:

$$T(n) = T\left(\frac{n}{2} - \frac{1}{2}\right) + c = \left[T\left(\frac{n}{4} - \frac{1}{2}\right) + c\right] + c = T\left(\frac{n}{4} - \frac{1}{2}\right) + 2c = \left[T\left(\frac{n}{8} - \frac{1}{2}\right) + c\right] + 2c = T\left(\frac{n}{8} - \frac{1}{2}\right) + 3c = \cdots = T\left(\frac{n}{2^i} - \frac{1}{2}\right) + ic$$

לאחר $i = \log\left(\frac{2}{3}n\right)$ צעדים נקבל כי:

$$T(n) = T\left(\frac{n}{2^i} - \frac{1}{2}\right) + ic = \mathbf{\Theta}(1) + c * \log\left(\frac{2}{3}n\right) = \mathbf{\Theta}(\log(n))$$

בסה"כ בשני המקרים קיבלנו כי :

$$T(n) = \mathbf{\Theta}(\log(n))$$

שאלה 5

א. על מנת לבנות אלגוריתם אשר יעמוד בתנאים הנ"ל נעזר באלגוריתם BinarySearch אותו פגשנו פעמים רבות בעבר: אלגוריתם אשר מקבל מערך ממוין (A), אינדקס התחלתי (start), אינדקס סופי (end) ואיבר אותו אנו מחפשים (X), ומחזיר את האינדקס בו נמצא X (1- במידה ולא קיים). ידוע כי זמן הריצה של האלגוריתם הנ"ל הינו O(log(end-start)).

כעת נכתוב את האלגוריתם אותו אנו מחפשים(בקריאה הראשונה i=2):

Function find(i, X, A)

If(i>A.length) return **BinarySearch** (i/2, A.length, X, A)

If(A[i]=X) return i

If(A[i]<X) **find**(i*2, X, A)

else return **BinarySearch** (i/2, i, X, A)

<u>תיאור קצר של האלגוריתם:</u> הפונקציה הרקורסיבית מקבלת אינדקס i מסוים שהוא חזקה של 2 (בפעם הראשונה 2), ובודקת היכן נמצא X במערך ביחס לאינדקס. כמובן שאם X נמצא במקום ה i היא תחזיר את i, במידה והאיבר נמצא "מימינה" ל i הפונקציה קוראת לעצמה עם i*2. במידה והאינדקס שקיבלה חורג מגודל המערך, משמע ש X נמצא בין i/2 לאיבר האחרון במערך ואת החיפוש הנ"ל יבצע BinarySearch. במידה ולא ובכל זאת X נמצא משמאל ל i, האיבר נמצא בין i/2 ל i וגם הפעם ניתן ל BinarySearch לבצע את החיפוש עבורנו.

<u>ניתוח זמן ריצה</u>: נשים לב כי הקריאה ל BinarySearch מתבצעת לכל היותר פעם אחת בלבד לאחר ש i גדול מהאינדקס של X. בזמן הקריאה לפונקציה BinarySearch קל לראות כי end-start הוא לכל היותר d, משום ש end הוא לכל היותר פעמיים start, אז end-start הוא לכל היותר start ו start הוא לכל היותר d. לכן: $\log(end-start) = \boldsymbol{o}(\log d)$. בנוסף, זמן הריצה של הפונקציה הרקורסיבית הוא log(d)+c משום שהפונקציה קוראת לעצמה log(d) פעמים עד שהגענו ל X או עברנו אותו .
ולסיכום הנוסחה לחישוב זמן הריצה הינה:

$$T(n) = f(d) + BS(d) = \log(d) + c + \boldsymbol{O}(\log(d)) = \boldsymbol{O}(\log(d))$$

ב.

```
Function median(A, B, C, D)
    L = (A.length+B.length+C.length+D.length)/2
    iA, iB, iC, iD ← 0
    A.add, B.add, C.add, D.add ←∞
    output
    while(L>=0)
    if (A[iA]<=B[iB] & A[iA]<=C[iC] & A[iA]<=D[iD])
        output=A[iA], iA++
    else if (B[iB]<=A[iA] & B[iB]<=C[iC] & B[iB]<=D[iD])
            output=B[iB], iB++
         else  if (C[iC]<=A[iA] & C[iC]<=B[iB] & C[iC]<=D[iD])
                output=C[iC], iC++
              else   if (D[iD]<=A[iA] & D[iD]<=B[iB] & D[iD]<=C[iC])
                    output=D[iD], iD++
    L=L-1
return output
```

הפונקציה בעלת איטרציה בודדת והיא הקובעת את זמן הריצה של הפונקציה. האיטרציה פועלת n/2 פעמים, והקבועים בכל ריצה אינם משפיעים על זמן הריצה הכולל ולכן קיבלנו בסה"כ כי:

$$T(n) = \boldsymbol{O}(n)$$