עבודת בית 1

שאלה 1:

שלב ראשון

צ״ל מיהי $g(n)$ כך ש $f_i(n) = \theta(g(n))$.

כלומר צריך למצוא $c_1, c_2 > 0$ כך ש $0 \leq c_1 * g(n) \leq f(n) \leq c_2 * g(n)$ לכל $n \geq n_0$

$1 - \ f_1(n) = 2022. \qquad g(n) = 1. \qquad f_1(n) \in \theta(g(n))$

$$0 \leq c_1 * 1 \leq 2022 \leq c_2 * 1$$

$$0 \leq c_1 \leq 2022 \leq c_2$$

נבחר $c_1 = 1, c_2 = 2023, n_0 = 1$

$2 - \ f_2(n) = \log(n^8). \qquad g(n) = \log(n). \qquad f_2(n) \in \theta(g(n))$

$$0 \leq c_1 * \log(n) \leq 8 * \log(n) \leq c_2 * \log(n) \quad /: \log(n)$$

$$0 \leq c_1 \leq 8 \leq c_2$$

נבחר $c_1 = 1, c_2 = 9, n_0 = 1$

$3 - \ f_3(n) = 2^{\sqrt{n}}. \qquad g(n) = 2^{\sqrt{n}}. \qquad f_3(n) \in \theta(g(n))$

$$0 \leq c_1 * 2^{\sqrt{n}} \leq 2^{\sqrt{n}} \leq c_2 * 2^{\sqrt{n}} \quad /: 2^{\sqrt{n}}$$

$$0 \leq c_1 \leq 1 \leq c_2$$

נבחר $c_1 = 1, c_2 = 2, n_0 = 1$

$$4 - \ f_4(n) = 3n^3 + 2 * \log(n) + 1. \qquad g(n) = n^3. \qquad f_4(n) \in \theta(g(n))$$

$$0 \le \mathrm{c}_1 * n^3 \le 3n^3 + 2 * \log(n) + 1 \le c_2 * n^3 \quad /: n^3$$

$$0 \le \mathrm{c}_1 \le 3 + \frac{2 * \log(n)}{n^3} + \frac{1}{n^3} \le c_2$$

נבחר $c_1 = 3, c_2 = 5, n_0 = 2$

$$5 - f_5(n) = 4^{2^n} \qquad g(n) = 4^{2^n} \qquad f_5(n) \in \theta(g(n))$$

$$0 \le \mathrm{c}_1 * 4^{2^n} \le 4^{2^n} \le c_2 * 4^{2^n} \quad /: 4^{2^n}$$
$$0 \le \mathrm{c}_1 \le 1 \le c_2$$

נבחר $c_1 = 1, c_2 = 1, n_0 = 1$

$$6 - f_6(n) = \frac{1}{\sqrt{n}} \qquad g(n) = \frac{1}{\sqrt{n}} \qquad f_6(n) \in \theta(g(n))$$
$$0 \le c_1 * \frac{1}{\sqrt{n}} \le \frac{1}{\sqrt{n}} \le c_2 * \frac{1}{\sqrt{n}} \quad :\backslash \frac{1}{\sqrt{n}}$$

$$0 \le c_1 \le 1 \le c_2$$
נבחר $c_1 = 1, c_2 = 1, n_0 = 1$

$$7 - f_7(n) = 6^{\log_{\sqrt{6}}(n)} = n^{\log_{6^{\frac{1}{2}}}(6)} = n^2 \qquad g(n) = n^2 \qquad f_7(n) \in \theta(g(n))$$

$$0 \le c_1 * n^2 \le n^2 \le c_2 * n^2 \quad :/n^2$$

$$0 \le c_1 \le 1 \le c_2$$

נבחר $c_1 = 1, c_2 = 1, n_0 = 1$

$8 - f_8(n) = 2^{64}$ $\qquad g(n) = 1$ $\qquad f_8(n) \in \theta(g(n))$

$$0 \le c_1 \le 2^{64} \le c_2$$

נבחר $c_1, c_2 = 2^{64}, n_0 = 1$

$9 - f_9(n) = 2^{4^n}$ $\qquad g(n) = 2^{4^n}$ $\qquad f_9(n) \in \theta(g(n))$

$$0 \le c_1 * \ 2^{4^n} \ \le 2^{4^n} \le c_2 * 2^{4^n} \quad :/2^{4^n}$$

$$0 \le c_1 \le 1 \le c_2$$

נבחר $c_1, c_2 = 1, n_0 = 1$

$10 - f_{10}(n) = n^n$ $\qquad g(n) = n^n$ $\qquad f_{10}(n) \in \theta(g(n))$

$$0 \le c_1 * \ n^n \ \le n^n \le c_2 * n^n \quad :/n^n$$

$$0 \le c_1 \le 1 \le c_2$$

נבחר $c_1, c_2 = 1, n_0 = 1$

$11 - f_{11}(n) = \log{(2^n * n^2)}$ $\qquad g(n) = \mathrm{n}$ $\qquad f_{11}(n) \in \theta(g(n))$

$$0 \le c_1 \le \log(2^n * n^2) = \log 2^n + \log n^2 \le c_2$$

$$0 \le c_1 \le n + 2\log n \le c_2 \quad :/n$$

$$0 \le c_1 \le 1 + \frac{2\log n}{n} \le c_2$$

נבחר $c_1 = 1, c_2 = 3, n_0 = 2$

$12 - f_{12}(n) = \log\left(n^{\frac{1}{2}}\right) = \dfrac{\log n}{2}$ $\qquad g(n) = \log n$ $\qquad f_{12}(n) \in \theta(g(n))$

$$0 \le c_1 * \log n \ \le \frac{\log n}{2} \le c_2 * \log n \quad :/\log n$$

$$0 \le c_1 \le \frac{1}{2} \le c_2$$

$c_1, c_2 = \frac{1}{2}, n_0 = 1$ נבחר

$13 - f_{13}(n) = 4^{\mathrm{n}}$ $\quad g(n) = 4^{\mathrm{n}}$ $\quad f_{13}(n) \in \theta(g(n))$

$$0 \le c_1 * 4^{\mathrm{n}} \le 4^{\mathrm{n}} \le c_2 * 4^{\mathrm{n}} \quad :/4^{\mathrm{n}}$$

$$0 \le c_1 \le 1 \le c_2$$

$c_1, c_2 = 1, n_0 = 1$ נבחר

$14 - f_{14}(n) = \frac{2\mathrm{n}}{7}$ $\quad g(n) = \mathrm{n}$ $\quad f_{14}(n) \in \theta(g(n))$

$$0 \le c_1 * \mathrm{n} \le \frac{2\mathrm{n}}{7} \le c_2 * \mathrm{n} \quad :/\mathrm{n}$$

$$0 \le c_1 \le \frac{2}{7} \le c_2$$

$c_1 = \frac{1}{7}, c_2 = 1, n_0 = 1$ נבחר

$15 - f_{15}(n) = \frac{1}{3\mathrm{n}^2}$ $\quad g(n) = \frac{1}{3\mathrm{n}^2}$ $\quad f_{15}(n) \in \theta(g(n))$

$$0 \le c_1 * \frac{1}{3\mathrm{n}^2} \le \frac{1}{3\mathrm{n}^2} \le c_2 * \mathrm{n} \quad :/\frac{1}{3\mathrm{n}^2}$$

$$0 \le c_1 \le 1 \le c_2$$

$c_1, c_2 = 1, n_0 = 1$ נבחר

שלב שני :

צריך לציין אילו פונקציות מקיימות $f_i(n) = \Theta(f_k(n))$

$f_i(n) = \Theta(f_k(n))$ ולפי מה שלמדנו בכיתה ש מסימטריות מתקיים

$f_k(n) = \Theta(f_i(n))$ ולכן מספיק שנראה רק כיוון אחד והשני יהיה בה"כ.

1 – $f_1(n) = 2022$ $\qquad f_8(n) = 2^{64}$

הראינו בשלב קודם כי $f_1(n), f_8(n) = \Theta(1)$ ומכאן נובע כי $f_1(n) = f_8(n)$

$f_8(n) = \Theta\big(f_1(n)\big) \quad \cap \quad f_1(n) = \Theta\big(f_8(n)\big) \quad \rightarrow \quad f_1(n) = f_8(n)$ מ.ש.ל

2 – $f_2(n) = \log(n^8)$ $\qquad f_{12}(n) = \log\left(n^{\frac{1}{2}}\right)$

הראינו בשלב ראשון כי $f_{12}(n), f_2(n) = \Theta(\log n)$ ומכאן נובע כי $f_{12}(n) = f_2(n)$

$f_2(n) = \Theta\big(f_{12}(n)\big) \quad \cap \quad f_{12}(n) = \Theta\big(f_2(n)\big) \qquad f_{12}(n) = f_2(n)$ מ.ש.ל

3 – $f_4(n) = \log(2^n * n^2)$ $\qquad f_{14}(n) = \frac{2n}{7}$

הראינו בשלב ראשון כי $f_{14}(n), f_4(n) = \Theta(n)$ ומכאן נובע כי $f_{14}(n) = f_4(n)$

$f_4(n) = \Theta\big(f_{14}(n)\big) \quad \cap \quad f_{14}(n) = \Theta\big(f_4(n)\big) \qquad f_{14}(n) = f_4(n)$ מ.ש.ל

שלב שלישי:

הפונקציות	סדר גודל אסימפטומטי (מהקטן לגדול)
1) $f_{15}(n) = \frac{1}{3n^2}$	$\boldsymbol{\Theta}(\frac{\mathbf{1}}{\mathbf{3}\boldsymbol{n}^{\mathbf{2}}})$
2) $f_6(n) = \frac{1}{\sqrt{n}}$	$\boldsymbol{\Theta}\left(\frac{\mathbf{1}}{\sqrt{\boldsymbol{n}}}\right)$
3) $f_1(n) = 2022\ , f_8(n) = 2^{64}$	$\boldsymbol{\Theta}(\mathbf{1})$
4) $f_2(n) = \log n^8\ ,$ $f_{12}(n) = \log n^{\frac{1}{2}}$	$\boldsymbol{\Theta}(\mathbf{log}\,\boldsymbol{n})$
5)$f_{14}\left(\frac{2n}{7}\right), f_{11}(n) = \log(2^n n^2)$	$\boldsymbol{\Theta}(\mathbf{n})$
6) $f_7(n) = 6^{\log_{\sqrt{6}}(n)}$	$\boldsymbol{\Theta}(\mathbf{n^2})$
7) $f_4(n) = 3n^3 + 2\log(n) + 1$	$\boldsymbol{\Theta}(\mathbf{n^3})$
8) $f_3(n) = 2^{\sqrt{n}}$	$\boldsymbol{\Theta}\left(\mathbf{2}^{\sqrt{\mathbf{n}}}\right)$
9) $f_{13}(n) = 4^n$	$\boldsymbol{\Theta}(\mathbf{4}^{\boldsymbol{n}})$
10) $f_{10}(n) = n^n$	$\boldsymbol{\Theta}(\boldsymbol{n}^{\boldsymbol{n}})$
11) $f_5(n) = 4^{2^4}$	$\boldsymbol{\Theta}\left(\mathbf{4}^{\mathbf{2}^{\mathbf{4}}}\right)$
12) $f_9(n) = 2^{4^n}$	$\boldsymbol{\Theta}\left(\mathbf{2}^{\mathbf{4}^{\boldsymbol{n}}}\right)$

כעת נוכיח את סדר הפונקציות בטבלה (שכל שורה מהווה חסם עליון לשורה שלפניה).

כלומר לפי ההגדרה של $O(g)$ צ״ל שקיימים קבועים c, n_0 כך ש עבור כל $n_0 \leq n$ מתקיים $0 \leq$ $g_i(n) \leq c * g_{i+1}(n)$

נסמן את סדר הגודל של הפונקציות ב $g(i)$ כך ש i מהווה מספר השורות בטבחה.

1. $$g_1(n) = \frac{1}{3n^2} = O\big(g_2(n)\big) = O\left(\frac{1}{\sqrt{n}}\right)$$

$$0 \leq \frac{1}{3n^2} \leq c * \frac{1}{\sqrt{n}} \qquad :/\frac{1}{\sqrt{n}}$$

$$0 \leq \frac{1}{3n^{\frac{3}{2}}} \leq c$$

מכאן $c = 1, n_0 = 1$ ולכן $g_1(n) \epsilon O(g_2(n))$

2. $$g_2(n) = \frac{1}{\sqrt{n}} = O\big(g_3(n)\big) = O(1)$$

$$0 \leq \frac{1}{\sqrt{n}} \leq c * 1$$

מכאן $c = 1, n_0 = 1$ ולכן $g_2(n) \epsilon O(g_3(n))$

3. $$g_3(n) = 1 = O\big(g_4(n)\big) = O(\log n)$$

$$0 \leq 1 \leq c * \log n \qquad :/\log n$$

$$0 \leq \frac{1}{\log n} \leq c$$

מכאן $c = 1, n_0 = 2$ ולכן $g_3(n) \epsilon O(g_4(n))$

4. $$g_4(n) = \log n = O\big(g_5(n)\big) = O(n)$$

$$0 \leq \log n \leq c * n \qquad :/n$$

$$0 \leq \frac{\log n}{n} \leq c$$

מכאן $c = 1, n_0 = 2$ ולכן $g_4(n) \epsilon O(g_5(n))$

5. $g_5(n) = n = O\big(g_6(n)\big) = O(n^2)$

$$0 \le n \le c * n^2 \quad :/n^2$$

$$0 \le \frac{1}{n} \le c$$

מכאן $c = 1, n_0 = 1$ ולכן $g_5(n) \epsilon O(g_6(n))$

6. $g_6(n) = n^2 = O\big(g_7(n)\big) = O(n^3)$

$$0 \le n^2 \le c * n^3 \quad :/n^3$$

$$0 \le \frac{1}{n} \le c$$

מכאן $c = 1, n_0 = 1$ ולכן $g_6(n) \epsilon O(g_7(n))$

7. $g_7(n) = n^3 = O\big(g_8(n)\big) = O\big(2^{\sqrt{n}}\big)$

$$0 \le n^3 \le c * 2^{\sqrt{n}} \quad :/2^{\sqrt{n}}$$

$$0 \le \frac{n^3}{2^{\sqrt{n}}} \le c$$

מכאן נסיק כי כמעט החל מ $g_7(n) \epsilon O(g_8(n)), c = 1, n_0 = 854$

8. $g_8(n) = 2^{\sqrt{n}} = O\big(g_9(n)\big) = O(4^n)$

$$0 \le 2^{\sqrt{n}} \le c * 4^n \quad :/4^n$$

$$0 \le \frac{2^{\sqrt{n}}}{4^n} \le c$$

מכאן $c = 1, n_0 = 1$ ולכן $g_8(n) \epsilon O(g_9(n))$

9. $g_9(n) = 4^n = O\big(g_{10}(n)\big) = O(n^n)$

$$0 \le 4^n \le c * 4^n \quad :/n^n$$

$$0 \le \frac{4^n}{n^n} = \left(\frac{4}{n}\right)^n \le c$$

מכאן $c = 1, n_0 = 0$ ולכן $g_9(n) \epsilon O(g_{10}(n))$

10. $g_{10}(n) = n^n = O\big(g_{11}(n)\big) = O\big(4^{2^n}\big)$

$$0 \le n^n \le c * 4^{2^n} \quad :/4^{2^n}$$

$$0 \le \frac{n^n}{4^{2^n}} = \left(\frac{n}{4^2}\right)^n \le c$$

מכאן $c = 1, n_0 = 1$ ולכן $g_{10}(n) \epsilon O(g_{11}(n))$

11. $g_{11}(n) = 4^{2^n} = O\big(g_{12}(n)\big) = O\big(2^{4^n}\big)$

$$0 \le 4^{2^n} \le c * 2^{4^n} \quad :/2^{4^n}$$

$$0 \le \frac{4^{2^n}}{2^{4^n}} = \left(\frac{2^4}{2^4}\right)^n = 1 \ \le c$$

מכאן $c = 1, n_0 = 1$ ולכן $g_{11}(n) \epsilon O(g_{12}(n))$

שאלה 2:

1. נניח $f(n)=\theta(g(n))$ וגם $g(n)=\theta(h(n))$ צ"ל $f(n)=\theta(h(n))$

$f(n)=\theta(g(n))$ אזי קיימים $c_1,c_2>0$ המקיימים $0\le c_1*g(n)\le f(n)\le c_2*g(n)$ לכל $n_0\le n$

$g(n)=\theta(h(n))$ אזי קיימים $c_3,c_4>0$ המקיימים $0\le c_3*h(n)\le g(n)\le c_4*h(n)$ לכל $n_0\le n$

$$0\le c_3*h(n)\le g(n) \quad :/*c_1$$

$$0\le c_3*c_1*h(n)\le c_1*g(n)\le f(n)$$

$$g(n)\le c_4*h(n) \quad :/c_2$$

$$f(n)\le g(n)*c_2\le c_4*c_2*h(n)$$

$$c_1*c_3*h(n)\le f(n)\le h(n)*c_2*c_4$$

מכאן קיימים $b=c_1*c_3$, $a=c_2*c_4$ קבועים עבורם מתקיים $n\ge n_0$ כך ש $b*h(n)\le f(n)\le a*h(n)$ ולכן $f(n)=\theta(h(n))$.

2. מספיק להראות כי $f(n)=\theta(f(n))$ אזי באופן מידי

$$f(n)=O\big(f(n)\big), f(n)=\Omega(f(n))$$

כלומר צ"ל כי קיימים $c_1,c_2>0$ כך ש $0\le c_1*f(n)\le f(n)\le c_2*f(n)$ לכל $n_0\le n$

$$0\le c_1*f(n)\le f(n)\le c_2*f(n) \quad :\backslash f(n)$$

$$0\le c_1\le 1\le c_2$$

נבחר $c_1,c_2=1$ $n_0=1$ ולכן $f(n)=\theta(f(n))$

3. נוכיח סימטריות
נראה הכלה דו כיוונית

$$g(n) = \theta\big(f(n)\big) - -\!\!-\!\!\rightarrow f(n) = \theta\big(g(n)\big) \qquad :\longrightarrow$$

נניח כי $g(n) = \theta\big(f(n)\big)$ צ״ל $f(n) = \theta\big(g(n)\big)$

$g(n) = \theta\big(f(n)\big)$ אזי קיימים $c_1, c_2 >= 0$ כך ש $0 \le f(n) * c_1 \le g(n) \le c_2 * f(n)$

$$0 \le f(n) * c_1 \le g(n) \ \ :\backslash c_1$$

$$0 \le f(n) \le \frac{1}{c_1} * g(n)$$

$$g(n) \le c_2 * f(n) \quad :\backslash c_2$$

$$\frac{1}{c_2} * g(n) \le f(n)$$

$$\frac{1}{c_2} * g(n) \le f(n) \ \le \frac{1}{c_1} * g(n)$$

ולכן מצאנו שני קבועים $\frac{1}{c_2}, \frac{1}{c_1}$ כך ש $0 \le \frac{1}{c_2} * g(n) \le f(n) \ \le \frac{1}{c_1} * g(n)$
לכל $n_0 \le n$ ומכאן נסיק כי $f(n) = \theta\big(g(n)\big)$

$$f(n) = \theta\big(g(n)\big) - -\!\!-\!\!\rightarrow g(n) = \theta\big(f(n)\big) \qquad :\leftarrow$$

נניח כי $f(n) = \theta\big(g(n)\big)$ צ״ל $g(n) = \theta\big(f(n)\big)$

$f(n) = \theta\big(g(n)\big)$ אזי קיימים $c_1, c_2 >= 0$ כך ש $0 \le g(n) * c_1 \le f(n) \le c_2 * g(n)$

$$0 \le g(n) * c_1 \le f(n) \ \ :\backslash c_1$$

$$0 \le g(n) \le \frac{1}{c_1} * f(n)$$

$$f(n) \le c_2 * g(n) \quad :\backslash c_2$$

$$\frac{1}{c_2} * f(n) \le g(n)$$

$$\frac{1}{c_2} * f(n) \le g(n) \ \le \frac{1}{c_1} * f(n)$$

ולכן מצאנו שני קבועים $\frac{1}{c_2}, \frac{1}{c_1}$ כך ש $0 \le \frac{1}{c_2} * f(n) \le g(n) \ \le \frac{1}{c_1} * f(n)$

לכל $n_0 \le n$ ומכאן נסיק כי $g(n) = \theta(f(n))$

4. נוכיח אנטי-סימטריות

נראה הכלה דו כיוונית

$\longrightarrow$: $\qquad f(n) = O(g(n)) ---\!\!\rightarrow g(n) = \Omega(f(n))$

נניח כי $f(n) = O(g(n))$ צ"ל $g(n) = \Omega(f(n))$

$f(n) = O(g(n))$ אזי קיים $c_1 >= 0$ כך ש $0 \le f(n) \le c_1 * g(n)$

$$0 \le f(n) \le c_1 * g(n) \qquad :/c_1$$

$$0 \le \frac{1}{c_1} * f(n) \le g(n)$$

ולכן מצאנו קבוע $\frac{1}{c_1}$ כך ש $g(n) = \Omega(f(n))$

$\longleftarrow$: $\qquad g(n) = \Omega(f(n)) ---\!\!\rightarrow f(n) = O(g(n))$

נניח כי $g(n) = \Omega(f(n))$ צ"ל $f(n) = O(g(n))$

$g(n) = \Omega(f(n))$ אזי קיים $c_1 >= 0$ כך ש $0 \le c_1 * g(n) \le f(n)$

$$0 \le c_1 * g(n) \le f(n) \qquad :/c_1$$

$$0 \le g(n) \le \frac{1}{c_1} * f(n)$$

ולכן מצאנו קבוע $\frac{1}{c_1}$ כך ש $f(n) = O(g(n))$

5. יהיו $p_1(n), p_2(n)$ פולינומים בחזקה n_1, n_2 בהתאמה.

$$p_1(n) = n^{n_1} \in \theta(n^{n_1})$$

$$p_2(n) = n^{n_2} \in \theta(n^{n_2})$$

צריך למצוא פונקציה $g(n)$ כך שמתקיים $p_1\left(p_2(n)\right) = \theta(g(n))$

$$p_1\left(p_2(n)\right) = p_1(n^{n_2}) = (n^{n_2})^{n_1} = n^{n_2 * n_1} \in \theta(n^{n_2 * n_1})$$

שאלה 3:

צריך למצוא חסם אסימפטוטי הדוק θ עבור $T(n)$.

ננית כי $T(n)$ קבועה עבור n קבוע.

א. $T(n) = T(\sqrt{n}) + 1$

<u>נשתמש בשיטת האיטרציה</u>

$$T(n) = T(\sqrt{n}) + 1 = T\left(n^{\frac{1}{2}}\right) + 1 =$$
$$\left(T\left(n^{\frac{1}{4}}\right) + 1\right) + 1 = T\left(n^{\frac{1}{4}}\right) + 2 =$$
$$\left(T\left(n^{\frac{1}{8}}\right) + 1\right) + 2 = T\left(n^{\frac{1}{2^3}}\right) + 3 =$$
$$= \cdots =$$

נזהה שבכל פעם חזקת ה2 תגדל באותו גודל של העבודה הנלווית, אזי נסמן הנוסחה כך

$$T(n) = T\left(n^{\frac{1}{2^i}}\right) + i$$

כעת ננסה למצוא את i בהתאם לתנאי העצירה.

נראה כי תאי העצירה שהוא $T(2)$, ולכן לפי ההנחה נסמן $T(2) = k$ כך ש k קבוע כלשהו.

$$n^{\frac{1}{2^i}} = 2$$

נפעיל לוג בשני האגפים

$$\log(n^{\frac{1}{2^i}}) = \log 2 \dashrightarrow \frac{\log n}{2^i} = 1 \rightarrow 2^i = \log n$$

נפעיל שוב לוג בשני האגפים

$$\log(\log n) = i$$

כעת נציב בנוסחה

$$T(n) = T\left(n^{\frac{1}{2^i}}\right) + i = T\left(n^{\frac{1}{2^{\log(\log n)}}}\right) + \log(\log n) = T(2) + \log(\log n) =$$
$$= k + \log(\log n)$$

כיוון k קבוע שהוא שווה ל $\theta(1)$ וגם $\log(\log n) \in \theta(\log(\log n)$ הניחוש שלנו יהיה

$$T(n) = \theta(\log\log n)$$

כעת נוכיח את הניחוש שלנו באינדוקציה:

<u>מקרה בסיס $n = 4$</u>

$$0 \le c_1 * \log\log 4 \le T(4) \le c_2 * \log\log 4$$

$$0 \le c_1 * 1 \le T(4) \le c_2 * 1$$

נתון כי $T(n)$ קבועה, נבחר $c_1, c_2 = T(n)$

ולכן הראינו שתנאי הבסיס מתקיים.

<u>הנחת האינדוקציה:</u> נניח עבור כל $m < n$, $T(m) = \theta(\log\log m)$

כלומר $(0 \le c_1 * \log\log m \le T(m) \le c_2 * \log\log m)$

נוכיח עבור n כי $T(n) = \theta(\log\log n)$

נחלק לשני מקרים:

← <u>נראה חסם תחתון</u> (צ"ל שקיים קבוע c_1 כך ש $0 \le c_1 * \log(\log(\sqrt{n})) \le T(n)$)

$$n^{\frac{1}{2}} < n$$

לפי הנחת האינדוקציה $0 \le c_1 * \log\log n^{\frac{1}{2}} \le T\left(n^{\frac{1}{2}}\right)$ נוסיף 1+ לביטוי ויישאר תקין.

$$0 \le c_1 * \log\log n^{\frac{1}{2}} + 1 = \log(\frac{1}{2} * \log n) + 1 \le T\left(n^{\frac{1}{2}}\right) + 1 = T(n)$$

$$0 \le c_1 * \log(\frac{1}{2} * \log n) + 1 = \log\frac{1}{2} + \log(\log n) + 1 \le T(n)$$

$$0 \le c_1 * (\log\frac{1}{2} + \log(\log n)) + 1 = \ c_1 * \log\log n - \ c_1 + 1 \le T(n)$$

$$0 \le c_1 * \log\log n \le c_1 * \log\log n + 1 - c_1 \le T(n)$$

המסומן באדום זה נתון

כעת נראה איזה קבוע חיובי צריך להיות c_1 כך ש יקיים את הביטוי

$$0 \le 1 - c_1 \rightarrow c_1 \le 1$$

מכאן קיים קבוע חיובי המקיים את ההגדרה ולכן $T(n) = \Omega(\log\log n)$

→ <u>נראה חסם עליון</u> (צ"ל שקיים קבוע c_2 כך ש $0 \le T(n) \le c_2 * \log(\log(\sqrt{n})$)

$$n^{\frac{1}{2}} < n$$

לפי הנחת האינדוקציה $0 \le T(n) \le c_2 * \log\log n^{\frac{1}{2}}$ נוסיף 1+ לביטוי ויישאר תקין.

$$0 \le T\left(n^{\frac{1}{2}}\right) + 1 = T(n) \le c_2 * \log\log n^{\frac{1}{2}} \ + 1 = \log(\frac{1}{2} * \log n) + 1$$

$$0 \le T(n) \le c_2 * \log\log n \ - c_2 + 1 \le c_2 * \log\log n$$

המסומן באדום זה נתון

כעת נראה איזה קבוע חיובי צריך להיות c_2 כך ש יקיים את הביטוי

$$0 \ge 1 - c_2 \rightarrow c_2 \ge 1$$

מכאן קיים קבוע חיובי המקיים את ההגדרה ולכן $T(n) = O(\log\log n)$

מכאן נסיק כי $T(n) = \theta(\log\log n)$ כנדרש.

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

<u>נשתמש בשיטת המאסטר</u>

$f(n) = n$ b=2 a=2

$n^{\log_a b} = n^{\log_2 2} = n$

נשים לב שהפונקציה $f(n)$ וגם $n^{\log_a b}$ באותו סדר גודל ולכן לפי כלל 2

T(n) $\in \theta(f(n)) \in \theta(n\log n)$

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

<u>נשתמש בשיטת המאסטר</u>

a=4 b=2 $f(n) = n^3 \log \mathrm{n}$

$g(n) = n^{\log_2 4} = n^2$

ניקח $\varepsilon = 1$, אז $n^{\log_b a+\varepsilon} = n^{\log_2 4+1} = n^3$

$0 \le n^3 * c \le n^3 \log \mathrm{n}$ *עבור* $n_0=1$, c=1 *לכל* $n>n_0$

לכן מתקיים $f(n) = \Omega(n^3)$ *עבור* $\varepsilon > 0$

נרצה להראות $af\left(\frac{n}{b}\right) < c * f(n)$ עבור c מסוים

$$4 * (\frac{n}{2})^2 * \log_2\left(\frac{n}{2}\right) < c * n^3\log n$$

$$\frac{n^3}{2} * (\log n - 1) < c * n^3 \log n$$

$$\left(\frac{1}{2} - \frac{1}{2\log n}\right) < c$$

נבחר $c=\frac{2}{3}$ ולכן לפי כלל 3 $T(n) \in \theta(f(n))$

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

<u>נשתמש בשיטת המאסטר</u>

$$a = 1\,, \qquad b = \frac{5}{2}, \qquad f(n) = 1$$

$$n^{\log_b a} = n^{(\log_{\frac{5}{2}} 1)} = n^0 = 1 = \theta(1)$$

נשים לב כי $f(n)$ וגם $\log_b a$ זהים ולכן לפי כלל 2 $T(n) = \theta(f(n) * \log n) = \theta(\log n)$.

ה. $T(n) = T(cn) + T\big((1-c)n\big) + 1 \quad 0 < c < 1$

<u>נשתמש בשיטת עץ הריקורסיה :</u>

ניתן להראות שמספר הרמות בעץ הריקורסיה הינו $\log_{\frac{1}{c}} n$

נסכום את כל הרמות ונגיע ל $T(n)=1+2+4+8+...+2^{log_{\frac{1}{c}} n} = n^{log_{\frac{1}{c}} 2}$

c<1 ולכן $n^{log_{\frac{1}{c}} 2} \leq n$ ולכן נרצה להראות $T(n) = \Theta(n)$

<u>נוכיח חסם תחתון :</u>

נוכיח באינדוקציה

מקרה בסיס עבור $n=1$:

$$0 \leq c1 * 1 \leq T(1)$$

$T(1)$ קבועה ולכן ניתן לבחור $c1$ כך ש $0 \leq c1 \leq T(1)$.

<u>הנחת האינדוקציה :</u>

נניח לכל $m<n$ מתקיים $T(m) = \Theta(m)$

$$T(cn) < T(n)$$

ולכן קיים c1 כך ש $0 \le c1 * cn \le cn$

$$T\big((1-c)n\big) < T(n)$$

ולכן קיים $c2$ כך ש $0 \le c2 * (1-c)n \le (1-c)n$

נבחר הקטן ביותר מבין $c1,c2$ נסמנו $c3$ ולכן כן מתקיים

$$0 \le c3cn + c3(1-c)n + 1 \le T(n) = T(cn) + T((1-c)n) + 1$$

$$c3cn + c3n - c3cn + 1 = c3n + 1 \le T(n)$$

נבחר $c3=1$ ואז הביטוי יתקיים

מכאן קיים קבוע חיובי המקיים את ההגדרה ולכן $T(n) = \Omega(n)$

<u>נוכיח חסם עליון:</u>

נזהה כי יש דמיון להוכחת חסם תחתון אבל את מה שניחשנו צריך להיות ניחוש יותר חזק אזי $0 \le T(n) \le c_2 * n - \partial$ עבור $\partial > 0$.

נשים לב כי אם $0 \le T(n) \le c_2 * n - \partial$ מתקיים אזי גם $0 \le T(n) \le c_2 * n -$

$\partial \le c_2 * n$ מתקיים.

<u>נוכיח באינדוקציה:</u>

מקרה בסיס : $n=1$

נתון כי $T(n)$ קבועה עבור n קבוע אזי מכאן נובע כי $0 \le T(1) + \partial \le c_2 * 1$

ולכן הראינו כי ניתן למצוא $c_2, \partial > 0$ המקיימים את ההגדרה.

<u>הנחת האינדוקציה:</u>

נניח לכל $m<n$ מתקיים $T(m) = \Theta(m)$

$$0 < c < 1$$

אזי קיים c2 כך ש $0 \le T(cn) \le c_2 cn - \partial$

מכאן גם $0 \le T((1-c)n) \le c_2(1-c)n - \partial$

אפשר להסיק עוד כי

$$0 \le T(n) = T(cn) + T\big((1-c)n\big) + 1 \le c_2 cn + c_2(1-c)n - 2\partial + 1$$

$$0 \le T(n) \le c_2 cn + c_2(1-c)n - 2\partial + 1 = c_2 n - 2\partial + 1$$

נזכיר כי אנו רוצים להראות קבוע חיובי המקיים

$$0 \le T(n) \le c_2 n - 2\partial + 1 \le c_2 n$$

מכאן צריך להתקיים

$$-2\partial+1\leq 0\rightarrow\frac{1}{2}\leq\partial$$

מכאן קיים קבוע חיובי המקיים את ההגדרה ולכן $T(n)=O(n)$

מכאן נסיק כי $T(n)=\theta(n)$ כנדרש.

ו. $T(n)=6T\left(\frac{1}{3}n\right)+n$

<u>נשתמש בשיטת המאסטר</u>

$$a=6\,,\qquad b=3,\qquad f(n)=n$$

$$n^{\log_b a}=n^{\log_3 6}=n^{1.631}$$

$$f(n)\in O(n^{1.6310-\varepsilon})\qquad 0<\varepsilon\leq 0.631$$

ולכן לפי כלל 1 $T(n)\in\theta(n^{1.631})$.

ז. $T(n)=T\left(\frac{1}{4}n\right)+T\left(\frac{3}{4}n\right)+n$

ראינו מקרה דומה בכיתה כי $T(n)=T\left(\frac{1}{4}n\right)+T\left(\frac{3}{4}n\right)+n\in\theta(n*\log n)$

אזי הניחוש שלנו הוא $T(n)=\theta(n*\log n)$.

<u>נוכיח את זה באינדוקציה</u>

מקרה בסיס $n=2$:

$$0\leq c_1*n*\log n\ \leq T(2)\leq c_2*n*\log n$$

$$0\leq c_1*2*\log 2\ \leq T(2)\leq c_2*2*\log 2$$

לפי הנחתינו $T(2)$ קבועה, ולכן מקבלים

$$0\leq c_1\leq\frac{T(2)}{2}\leq c_2$$

מכאן קיימים שני קבועים המקיימים את ההגדרה ולכן $T(n)=\theta(n*\log n)$.

הנחת האינדוקציה: נניח עבור כל $m<n$, $T(m)=\theta(n*\log m)$

כלומר $0\leq c_1*m*\log m\ \leq T(m)\leq c_2*m*\log m$

נוכיח עבור n כי $T(n)=\theta(n*\log n)$

נחלק לשני מקרים:

נראה חסם תחתון ← (צ"ל שקיים קבוע c_1 כך ש $0 \leq c_1 * n * \log n \ \leq T(n)$)

לפי הנחת האינדוקציה

$$0 \leq c_1 * \left(\frac{n}{4} * \log\left(\frac{n}{4}\right) + \frac{3n}{4} * \log\left(\frac{3n}{4}\right)\right) \leq T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right)$$

ניתן גם לרשום

$$0 \leq c_1 * \left(\frac{n}{4} * \log\left(\frac{n}{4}\right) + \frac{3n}{4} * \log\left(\frac{3n}{4}\right)\right) + n \ \leq T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n$$

$$0 \leq c_1 * \left(\frac{n}{4} * \log\left(\frac{n}{4}\right) + \frac{3n}{4} * \log\left(\frac{3n}{4}\right)\right) + n \leq T(n)$$

$$0 \leq c_1 * \frac{n}{4} * \log\left(\frac{n}{4}\right) + c_1 * \frac{3n}{4} * \log\left(\frac{3n}{4}\right) + \ n \leq T(n)$$

$$0 \leq c_1 * \frac{n}{4}(\log n - \log 4) + c_1 * \frac{3n}{4}(\log 3n - \log 4) + \ n \leq T(n)$$

$$0 \leq c_1 * \frac{n}{4}(\log n - 2) + c_1 * \frac{3n}{4}(\log 3 + \log n - 2) + \ n \leq T(n)$$

$$= c_1 * \frac{n}{4} * \log n - c_1 * \frac{n}{2} + c_1 * 1.185 * n + c_1 * \frac{3n}{4} * \log n - c_1 * \frac{3n}{2} + \ n$$

$$= c_1 * \left(\frac{n}{4} * \log n - \frac{n}{2} + 1.185 * n + \frac{3n}{4} * \log n - \frac{3n}{2}\right) + \ n$$

$$= c_1 * (-0.815 * n + n * \log n) + \ n \leq T(n)$$

$$= c_1 * n * \log n - c_1 * n * 0.815 + \ n \leq T(n)$$

נזכיר כי אני רוצה להראות $0 \leq c_1 * n \log n \leq T(n)$

נשים לב כי $c_1 * n \log n \leq c_1 * n * \log n - c_1 * n * 0.815 + \ n$

כדי שזה יתקיים אנו רוצים כי $n - c_1 * n * 0.815 \geq 0$

$$n - c_1 * n * 0.815 \geq 0 \rightarrow c_1 \leq 1.22$$

מכאן קיים קבוע חיובי המקיים את ההגדרה ולכן $T(n) = \Omega(n \log n)$

נראה חסם עליון ← (צ"ל שקיים קבוע c_1 כך ש $0 \leq T(n) \leq \ c_1 * n * \log n$)

לפי הנחת האינדוקציה

$$0 \leq T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) \leq c_1 * \left(\frac{n}{4} * \log\left(\frac{n}{4}\right) + \frac{3n}{4} * \log\left(\frac{3n}{4}\right)\right)$$

ניתן גם לרשום

$$0 \leq T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n \leq c_1 * \left(\frac{n}{4} * \log\left(\frac{n}{4}\right) + \frac{3n}{4} * \log\left(\frac{3n}{4}\right)\right) + n$$

$$0 \le T(n) \le c_1 * \left(\frac{n}{4} * \log\left(\frac{n}{4}\right) + \frac{3n}{4} * \log\left(\frac{3n}{4}\right)\right) + n$$

נפתח את הביטוי כמו קודם ונקבל

$$= 0 \le T(n) \le c_1 * n * \log n - c_1 * n * 0.815 + \ n$$

נזכיר כי אני רוצה להראות $0 \le T(n) \le c_1 * n \log n$

נשים לבּ כי $c_1 * n * \log n - c_1 * n * 0.815 + \ n \le c_1 * n \log n$

כדי שזה יתקיים אנו רוצים כי $n - c_1 * n * 0.815 \le 0$

$$n - c_1 * n * 0.815 \le 0 \rightarrow c_1 \ge 1.22$$

מכאן קיים קבוע חיובי המקיים את ההגדרה ולכן $T(n) = O(n \log n)$

מכאן נסיק כי $T(n) = \theta(n \log n)$ כנדרש.

שאלה 4:

a) **Function** **foo1 (A, key)**	סדר גודל
Index ← -1	$\theta(1)$
For i←0 **to** n-1	$\theta(n)$
If (A[i]==key)	$\theta(n)$
Index ←i	$\theta(1)$
Break	$\theta(1)$
Return index	$\theta(1)$

מכאן נסיק שהמקרה הגרוע ביותר הוא $\theta(n)$ כיוון ש key יכול להימצא בתא האחרון של המערך או לא יימצא בכלל ולכן סדר הגודל של הפונקציה $T(n) = \theta(n)$.

b) **Function** **Foo2 (A)**	סדר גודל
for I ← 0 to n-1	$\theta(n)$
for j ← n-1 to 0	$\theta(n^2)$
if (A[i] > A[j])	$\theta(n^2)$
A[i] = j	$\theta(n^2)$
else	$\theta(n^2)$
A[j] = i	$\theta(n^2)$

אנו רואים שישנם שתי forים נזהה מה בתוך הקוד ונראה שהמצב הגרוע ביותר הוא כאשר עוברים על המערך כולו וזה יקח זמן של $f(n) = \theta(n^2)$

c) **Function** **exp (base, n)**	סדר גודל
if (n = 0)	$\theta(1)$
return 1	$\theta(1)$
else if (n = 1)	$\theta(1)$

return base	$\theta(1)$
else	$\theta(1)$
return base · exp (base, n-1)	$T(n-1)$

$$T(n) = \begin{cases} \theta(1) & n = 1 \| n = 0 \\ T(n-1) + \theta(1) & n > 1 \end{cases}$$

כעת נראה ש $f(n) = \theta(n)$

<u>נשתמש בשיטת ההצבה</u> .

נוכיח באינדוקציה .

מקרה בסיס : n=0

$T(n) = \theta(1)$ אזי $T(n) = \theta(n)$

הנחת האינדוקציה נניח לכל $m<n$ מתקיים $T(m) = \theta(m)$

$n-1<n$ לכן מתקיים $T(n-1) = \theta(n-1)$

כלומר קיימים $c1,c2$ כך ש

$$0 \leq c1 * (n-1) \leq T(n-1) \leq c2 * (n-1)$$

אזי $T(n-1) = \theta(n)$.

$T(n)=T(n-1) + \theta(1)$ ולכן מתקיים $T(n) = \theta(n)$.

d) **<u>Function</u> exp2(base, n)**	<u>סדר גודל</u>
If(n=0)	$\theta(1)$
Return 1	$\theta(1)$
Else if (n=1)	$\theta(1)$
Return base	$\theta(1)$
Else if (mod(n,3) =0)	$\theta(1)$
Tmp←**exp2**(base, n/3)	$T(\frac{n}{3})$
Return tmp*tmp*tmp	$\theta(1)$
Else	$\theta(1)$

Return base*$\mathbf{exp2}$(base, n-1) $T(n-1)$

בקצרה הפונקציה פועלת כך שברגע n מתחלק ב 3 אזי מתבצעת קריאה רקוקסיבית עם הפרמטר n/3 ועוד מספר קבוע.
ברגע שלא מתחלק ב 3 אזי מתבצעת קריאה לאותה פונקציה עד ש ה n יגיע ויתחלק ב 3

אנו מקבלים מספר בבסיס כלשהו, נראה כי יש לנו 3 מקרים :

1. כאשר $n\%3 = 0$, נייצג את זמן הריצה כ T_0
2. כאשר $n\%3 = 1$, נייצג את זמן הריצה כ T_1
3. כאשר $n\%3 = 2$, נייצג את זמן הריצה כ T_2

מקרה 1 :

$$T_0(n) = T\left(\frac{n}{3}\right) + \theta(1) = T\left(\frac{n}{3}\right) + c_0$$

מקרה 2 :

$$T_1(n) = T_0(n-1) + \theta(1) = T_0(n-1) + c_1 = T\left(\frac{n-1}{3}\right) + c_0 + c_1$$

מקרה 3 :

$$T_2(n) = T_1(n-1) + \theta(1) = T_1(n-1) + c_2 =_{\text{מקרה 2}} T_0(n-2) + c_1 + c_2$$

$$=_{\text{מקרה 1}} T\left(\frac{n-2}{3}\right) + c_0 + c_1 + c_2$$

מהפיתוח הזה נסיק כי זמן הריצה תלוי ב T_0 ולכן ניתן לרשום את נוסחת הנסיגה המתארת את זמן הריצה באופן הבא :

$$T(n) = \begin{cases} \theta(1) & n = 0 \ or \ 1 \\ T\left(\frac{n}{3}\right) + \theta(1) & \text{אחרת} \end{cases}$$

כעת נפתור את נוסחת הנסיגה בשיטת האיטרציה ואחר כך ננחש פתרון ונוכיח אותו באינדוקציה.

$$T(n) = T\left(\frac{n}{3}\right) + c = \left(\frac{n}{9}\right) + 2c = \left(\frac{n}{27}\right) + 3c = \cdots = T\left(\frac{n}{3^i}\right) + ic$$

$$\frac{n}{3^i} = 1 ===> 3^i = n$$

$$\mathrm{i} * \log 3 = \log n$$

$$i = \frac{\log n}{\log 3} = \frac{1}{\log 3} * \log n$$

$$i = \log n$$

נציב בנוסחה ונקבל:

$$T(n) = T(1) + c\log n$$

הניחוש שלנו $T(n) = \theta(\log n)$

<u>נוכיח באינדוקציה:</u>

מקרה בסיס $n = 2$

$$0 \le c_1 * \log 2 \le T(2) \le c_2 * \log 2$$

$$0 \le c_1 \le T(2) \le c_2$$

$T(2)$ קבוע חיובי ולכן מקרה בסיס מתקיים.

הנחת האינדוקציה: נניח עבור כל $m < n$, $T(m) = \theta(\log m)$

כלומר $0 \le c_1 * \log m \le T(m) \le c_2 * \log m$

← <u>נראה חסם תחתון</u> (צ"ל שקיים קבוע c_1 כך ש $0 \le c_1 * \log n \le T(n)$)

$$T(n) = T\left(\frac{n}{3}\right) + c \geq_{\text{ה.א}} c_1 * \log\left(\frac{n}{3}\right) + c = c_1 * \log n - c_1 \log 3 + c$$

נבחר $c_1 = 0.3$, $n_0 = 1$ ולכן מתקבל כי $c_1 \log 3 < 1 \le c$ מכאן

$c_1 \log n - c_1 \log 3 + c \ge c_1 \log n$ ולבסוף $T(n) \ge c_1 \log n$

← <u>נראה חסם עליון</u> (צ"ל שקיים קבוע c_1 כך ש $0 \le T(n) \le c_1 * \log n$)

$$T(n) = T\left(\frac{n}{3}\right) + c \;.\; c\ \text{קבוע חיובי} \le_{\text{ה.א}} c_2 * \log\left(\frac{n}{3}\right) + c = c_2 * \log n - c_2 \log 3 + c$$

נבחר $c_1 = 1$, $n_0 = 1$ ולכן מתקבל כי $c_2 \log 3 > c$ מכאן

$c_2 \log n - c_2 \log 3 + c \le c_2 \log n + c$ ולבסוף $T(n) \le c_2 \log n$

e) **<u>Function</u>** expC (base, n)	<u>סדר גודל</u>
if (n = 0)	$\theta(1)$
return 1	$\theta(1)$
else if (n = 1)	$\theta(1)$
return base	$\theta(1)$
else if (mod (n, c) = 0)	$\theta(1)$
tmp ← expC (base, n/c)	$T(\frac{n}{c})$
ans ← 1	$\theta(1)$
for i ← 1 to c:	$\theta(c)$
ans ← ans * tmp	$\theta(c)$
return ans	$\theta(1)$
else	$\theta(1)$

return base · expC (base, n-1) $T(n-1)$

אנו מקבלים מספר בבסיס כלשהו, נראה כי יש לנו c מקרים:

1. כאשר $n\%c = 0$, נייצג את זמן הריצה כ T_0
2. כאשר $n\%c = 1$, נייצג את זמן הריצה כ T_1
3. כאשר $n\%c = 2$, נייצג את זמן הריצה כ T_2

נשים לב כי המקרה הגרוע ביותר הוא כש $n\%c = c-1$ ולכן נסמן המקרה הגרוע ביותר ב T_{c-1}

מקרה 1:

$$T_0(n) = T\left(\frac{n}{c}\right) + \theta(c) + \theta(1) = T\left(\frac{n}{c}\right) + k_c + c_0$$

מקרה 2:

$$T_1(n) = T_0(n-1) + \theta(1) = T_0(n-1) + c_1 = T\left(\frac{n-1}{c}\right) + c_0 + k_c + c_1$$

מקרה 3:

$$T_2(n) = T_1(n-1) + \theta(1) = T_1(n-1) + c_2 =_{\text{מקרה 2}} T_0(n-2) + c_1 + c_2$$

$$=_{\text{מקרה 1}} T\left(\frac{n-2}{c}\right) + c_0 + k_c + c_1 + c_2$$

כעת נראה מה יקרה כש $n\%c = c-1$

$$T_{c-1} = T_{c-2}(n-1) + \theta(1) = T_{c-2}(n-1) + \theta(1) + c_{c-1} = \cdots$$

$$= T_0\left(\frac{(n-c+1)}{c}\right) + c_0 + c_1 + c_2 + k_c + .. + c_{c-1}$$

מהפיתוח הזה נסיק כי זמן הריצה ניתן לרשומו באופן הבא:

$$T(n) = \begin{cases} \theta(1) & n = 0\ or\ 1 \\ T\left(\frac{n}{c}\right) + \theta(c) + \theta(1) & n\%c = 0 \\ T\left(\frac{(n-c+1)}{c}\right) + \theta(c) + \theta(1) & \text{אחרת} \end{cases}$$

נשים לב כי ישנם 2 מקרים ל c.

1. C קבוע כלשהו אזי

$$T(n) = \begin{cases} \theta(1) & n = 0\ or\ 1 \\ T\left(\frac{n}{c}\right) + \theta(1) & n\%c = 0 \\ T\left(\frac{(n-c+1)}{c}\right) + \theta(1) & \text{אחרת} \end{cases}$$

$$f(n) = \theta(1) \quad a = 1 \quad b = \frac{1}{c}$$

לכן לפי שיטת המאסטר כלל 2 $f(n) \in \theta\left(n^{\log_b a}\right) = \theta(n^0) = \theta(1)$

ולכן $T(n) \in \theta(\log n)$

2. C=n כלומר c תלוי ב n אזי

$$T(n) = \begin{cases} \theta(1) & n = 0 \ or \ 1 \\ T\left(\frac{n}{n}\right) + \theta(n) + \theta(1) & else \end{cases}$$

מכאן נסיק כי זמן הרצה כאשר c=n הוא $\theta(n)$

שאלה5:

a)

isSame (A, B)	
n←A.length	$\theta(1)$
create array AHelp[2n+1]	$\theta(1)$
create array BHelp[2n+1]	$\theta(1)$
for k←0 to 2n	$\theta(n)$
AHelp[k] ←0	$\theta(n)$
BHelp[k] ←0	$\theta(n)$
for i ←0 to n	$\theta(n)$
AHelp[A[i]] =AHelp[A[i]] +1	$\theta(n)$
for j ←0 to n	$\theta(n)$
BHelp[A[i]] =BHelp[A[i]] +1	$\theta(n)$
for m←0 to n	$\theta(n)$
if (AHelp[m]! =0 and BHelp[m]! =0)	$\theta(n)$
return false	$\theta(1)$
return true	$\theta(1)$

הסבר במילים :

נתחיל שני מערכים עזרה באורך של 2n+1 (כאשר האורך של A ו B הוא n) והערך בכל תא במערכים בהתחלה הוא 0

נעבור על כל מערך A ו B בנפרד ובכל פעם מופיע מספר כלשהוא מ 0 עד 2n נוסיף אחד במקום(אינדיקס) המספר הזה במערך העזרה שלו בהתאמה. וזה יקח $\theta(n)$

לדוגמה : אם המספר 5 מופיע פעמיים במערך B אז נוסיף אחד במערך העזרה BHelp בתא החמישי פעמיים.(זה גם סופיר כמה פעמים מופיע המספר חמש אז ניתן גם להציב המספר אחד וזה עובד גם)

לבסוף נעבור על שני מערכי העזרה ונראה אם באותו תא בשניהם ישנם יותר מ 0 אז האינדקס הזה מופיע בשניבם ויש להחזיר false אחרת נחזיר true .

ולפי כך קל לראות שסדר הגודל הגדול ביותר בקוד הזה הינו $\theta(n)$.

סעיף ב :

לפי הנחינות סעיף זה יצרנו פתרון דומה ל binary search כש זמן הריצה של הפתרון שלנו זהה לזה שב binary search.
הפתרון שלנו מיוצג כ כפיצות של 2 כדי לעבור על המערך בזמן יותר יעיל מכך לעבור על כולו, לשם כך בנינו שתי פונקציות.
הפונקציה הראשונה מקבלת את המערך ואת האיקס שלנו והיא בודקת בתנאי הראשון אם האיבר בתא הראשון במערך גדול מהאיקס שלנו או האיבר בתא האחרון במערך קטן מהאיקס שלנו אזי הפונקציה מיד תחזיר את הערך -1 כי האיקס לא נמצא בכלל במערך כיוון שהוא ממוין.
בתנאי השני והשלישי בודקים אם האיבר בתא האחרון או הראשון הוא האיקס עצמו ומחזירים את האינדקס שלו בהתאם.
ולבסוף קוראים לפונקצית העזר שלנו שהיא תמצא לנו את האינדקס.

```
Function searchMyKey (A, X)
        If (A [0]>x or A [A.size -1] <x)
                Return -1
        If (A [0] =X)
                Return 0
        If (A [A.size -1] =X)
                Return A.size-1
        Return helpSearchkey (1, A, X)
```

התנאי הראשון בפונקצית העזר בודק אם המערך באינדקס הזה שווה לאיקס ומחזירים את האינדקס.
בתנאי השני בודקים אם התא באינדקס הזה גדול מ איקס אזי תבדוק בעצם אם האינדקס הוא אורך כל התאים ואם זה נכון אזי האיקס שלנו לא נמצא ונחזיר -1, אחרת בודקים את התא הלפניה.
בתנאי השלישי נמנוע חריגה כי האם היו הקפיצות שלנו מעבר לאורך המערך אזי יש לנו בעיה ולכן ברגע שהקפיצה הבאה גדולה מאורך המערך אזי נתקדם בצעדים.
ולבסוף אם כל מה שלעיל לא מתקים אז נקפוץ בשתיים דרך חזרה לתנאי הפונקציה.

```
Function helpSearchkey (index, A, x)
```

```
If (A[index]=x)
        Return index
If (A[index]>x)
        If (index=A.size-1)
                Return -1
        Return helpSearchkey(index-1,A,x)
If (index*2>A.size-1)
        Return (index+1,A,x)
Return (index*2,A,x)
```

מכאן יש לנו פתרון לבעיית חיפוש key בעלת זמן ריצה של $\theta(\log d)$ כאשר d הוא מספר האיברים מופיעים לפני איבר x במערך בדומה לזמן ריצה של binary search שמפצלים את המערך.