שאלה 1

(1)

הפרכה : עבור $f_1(n) = \dfrac{1}{n}$

הכיוון "$f_1(n) = O(1) \Rightarrow f_1(n) = \Theta(1)$" לא מתקיים כיוון ש

$f_1(n) \neq \Omega(1)$, ולכן לפי הגדרת Θ, לא מתקיים $f_1(n) = \Theta(1)$.

נראה $f_1(n) \neq \Omega(1)$:

נניח בשלילה ש $f_1(n) = \Omega(1)$, לכן קיימים n_0וc כך שלכל $n > n_0$:

$c \leq f_1(n) = \dfrac{1}{n}$

בסתירה לכך ש $\dfrac{1}{n}$ שואף ל 0.

(2)

$$\sqrt{f_1(n)} = \Omega(f_1(n)) \iff f_1(n) = O(1)$$

הטענה נכונה.

$\Leftarrow$: נניח כי $f_1(n) = O(1)$. אז קיימים קבועים $c_1, n_0 > 0$ כך שלכל $n > n_0$,

$$f_1(n) \leq c_1$$

$$\Rightarrow \sqrt{f_1(n)} \leq \sqrt{c_1}$$

$$\Rightarrow \frac{f_1(n)}{\sqrt{f_1(n)}} \leq \frac{c_1}{\sqrt{c_1}}$$

$$\Rightarrow \sqrt{f_1(n)} \cdot c_1 \geq \sqrt{c_1} \cdot f_1(n)$$

$$\sqrt{f_1(n)} \geq \frac{\sqrt{c_1}}{c_1} \cdot f_1(n) \Rightarrow \sqrt{f_1(n)} \geq \frac{1}{\sqrt{c_1}} \cdot f_1(n)$$

נגדיר $c_2 = \frac{1}{\sqrt{c_1}}$. אז עבור הקבועים $n_0, c_2 > 0$, לכל $n > n_0$ מתקיים $\sqrt{f_1(n)} \geq c_2 \cdot f_1(n)$ ולכן $\sqrt{f_1(n)} = \Omega(f_1(n))$.

$\Rightarrow$: נניח כי $\sqrt{f_1(n)} = \Omega(f_1(n))$. אז קיימים קבועים $c_1, n_0 > 0$ כך שלכל $n > n_0$,

$$\sqrt{f_1(n)} \geq c_1 \cdot f_1(n)$$

$$\Rightarrow \sqrt{f_1(n)} \geq c_1 \cdot \sqrt{f_1(n)}\sqrt{f_1(n)}$$

מהגדרת תמונת f_1 נובע $\sqrt{f_1(n)} > 0$ ולכן:

$$\Rightarrow 1 \geq c_1 \cdot \sqrt{f_1(n)} \Rightarrow \sqrt{f_1(n)} \leq \frac{1}{c_1}$$

$$\Rightarrow f_1(n) \leq \frac{1}{c_1^2}$$

נגדיר $c_2 = \frac{1}{c_1^2}$ ולכן עבור הקבועים $n_0, c_2 > 0$, לכל $n > n_0$ מתקיים $f_1(n) \leq c_2$ ולכן $f_1(n) = O(1)$.

(3)

הפרכה:

עבור:

$f_2(n) = n$, $f_1(n) = n$

$g_2(n) = n^2$, $g_1(n) = n$

מתקיים

$f_1(n) = O(g_1(n))$ (כל פונקציה היא O של עצמה, עבור $c = 1$ וכל n_0)

ו-

$f_1(n) = O(g_1(n))$ ($n = O(n^2)$, הוכח בתרגול).

אבל,

$\frac{f_1(n)}{f_2(n)} = 1$ ו $\frac{g_1(n)}{g_2(n)} = \frac{1}{n}$

נראה ש $1 \neq O(\frac{1}{n})$:

נניח בשלילה שקיים c וn_0כל שלכל $n > n_0$:

$1 \leq \frac{c}{n} \Rightarrow n \leq c$

בסתירה לכך שאין חסם עליון לטבעיים.

(4)

$$\left| log\left(\frac{f_1}{g_1}\right) \right| = O(log(g_1)) \Rightarrow f_1(n) = O(g_1(n))$$

הטענה אינה נכונה. נתבונן לדוגמה בפונקציות $f_1(n) = n^2, g_1(n) = n$. מתקיים:

$$\left| log\left(\frac{n^2}{n}\right) \right| = |log\,(n)| \underbrace{=}_{n\geq 1} log(n)$$

מתקיים $log(n) = O(log(n))$ ולכן $\left| log\left(\frac{n^2}{n}\right) \right| = O(log(n))$, כלומר $\left| log\left(\frac{f_1}{g_1}\right) \right| = O(log(g_1))$.

נניח בשלילה כי $n^2 = O(n)$. אז קיימים c, n_0 כך שלכל $n \geq n_0$ מתקיים $n^2 \leq c \cdot n$. לכל $n \in \mathbb{N}$, מתקיים $n^2 \geq n$ ולכן בהכרח $c \geq 1$. כלומר $cn_0 \geq n_0$ ולכן עבור $n' = c \cdot \max\{n_0, 2\}$:

$$c^2 \cdot n'^2 \leq c^2 n' \Rightarrow n'^2 \leq n'$$

בסתירה לכך ש$1 > 2 \geq n'$. לכן $n^2 \neq O(n)$ כלומר $f_1 \neq O(g_1)$.

(5)

הוכחה:

נניח כי:

$$f_1(n) = \Theta(g_1(n))$$
$$f_2(n) = \Theta(g_2(n))$$

לכן

קיימים c_1,c_2 וn_1כך שלכל $n > n_1$:

$$c_1 g_1(n) \leq f_1(n) \leq c_2 g_1(n)$$

וגם

קיימים c_3,c_4 וn_2כך שלכל $n > n_2$:

$$c_3 g_2(n) \leq f_2(n) \leq c_4 g_2(n)$$

לכן

עבור n שגדול מ $n_3 = max\{n_1, n_2\}$:

$$\frac{c_1}{c_4}\frac{g_1(n)}{g_2(n)} \leq \frac{f_1(n)}{f_2(n)} \leq \frac{c_2}{c_3}\frac{g_1(n)}{g_2(n)}$$

עבור קבועים $c_5 = \frac{c_2}{c_3}$ ו $c_6 = \frac{c_1}{c_4}$ מתקיימת הגדרת $\frac{f_1(n)}{f_2(n)} = \Theta(\frac{g_1(n)}{g_2(n)})$.

(6)

אם $\left|log\left(\frac{f_1(n)}{g_1(n)}\right)\right| = O(f_2(n))$ וגם $\left|log\left(\frac{g_1(n)}{g_2(n)}\right)\right| = O(f_2(n))$ אז $\left|log\left(\frac{f_1(n)}{g_2(n)}\right)\right| = O(f_2(n))$.

הטענה נכונה. נתון כי קיימים $n_1, c_1 > 0$ כך ש:

$$\left|log\left(\frac{f_1(n)}{g_1(n)}\right)\right| \leq c_1 \cdot f_2(n)$$

מתכונות לוגריתמים נקבל:

$$\Rightarrow |log\left(f_1(n)\right) - log\left(g_1(n)\right)| \leq c_1 \cdot f_2(n)$$

נתון גם כי קיימים $n_2, c_2 > 0$ כך ש:

$$\left|log\left(\frac{g_1(n)}{g_2(n)}\right)\right| \leq c_2 \cdot f_2(n)$$

$$\Rightarrow |log\left(g_1(n)\right) - log\left(g_2(n)\right)| \leq c_2 \cdot f_2(n)$$

נגדיר $n_3 = \max\{n_1, n_2\}$. לכל $n \geq n_3$ כל אי השוויונות מתקיימים במקביל ולכן נוכל לחבר אותם:

$$|log\left(f_1(n)\right) - log\left(g_1(n)\right)| + |log\left(g_1(n)\right) - log\left(g_2(n)\right)| \leq c_1 \cdot f_2(n) + c_2 \cdot f_2(n)$$

מאי שוויון המשולש נקבל:

$$\Rightarrow \left|log\left(f_1(n)\right) - log\left(g_1(n)\right) + log\left(g_1(n)\right) - log\left(g_2(n)\right)\right| \le (c_1 + c_2) \cdot f_2(n)$$

נגדיר $c_3 = c_1 + c_2$. $c_2, c_1 > 0$ ולכן גם $c_3 > 0$.

$$\Rightarrow \left|log\left(f_1(n)\right) - log\left(g_2(n)\right)\right| \le c_3 \cdot f_2(n)$$

מתכונות לוגריתמים נקבל:

$$\Rightarrow \left|log\left(\frac{f_1(n)}{g_2(n)}\right)\right| \le c_3 \cdot f_2(n)$$

כלומר עבור הקבועים $c_3, n_3 > 0$ מתקיים $\left|log\left(\frac{f_1(n)}{g_2(n)}\right)\right| = O(f_2(n))$ כנדרש.

שאלה 2

טבלה:

(א)

$nlog^2(n)$:

n: עבור $n_0 = 2$ ו$c = 1$ מתקיים לכל $n \ge n_0$:

$$log(n) \ge 1 \Rightarrow log^2(n) \ge 1 \Rightarrow nlog^2(n) \ge 1 \cdot n$$

ולכן $nlog^2(n) = \Omega(n)$. נניח כי $nlog^2(n) = \Theta(n)$. אז $nlog^2(n) = O(n)$ ולכן קיימים $n_0, c > 0$ כך שלכל $n \ge n_0$:

$$nlog^2(n) \le c \cdot n \Rightarrow log^2(n) \le c \Rightarrow log(n) \le \sqrt{c}$$

לכן גם עבור $n' = \max\{2^{\sqrt{c}} + 1, n_0 + 1\}$ מתקיים

$$log(n') \le \sqrt{c}$$

$$\Rightarrow 2^{\sqrt{c}} + 1 \le n' \le 2^{\sqrt{c}}$$

בסתירה, ולכן $nlog^2(n) = \Omega(n)$ הוא היחס החזק ביותר בין הפונקציות.

$nlog(n)$: עבור $n_0 = 2$ ו $c = 1$ מתקיים לכל $n \geq n_0$:

$$log(n) \geq 1 \Rightarrow nlog(n) \cdot log(n) \geq 1 \cdot nlog(n) \Rightarrow nlog^2(n) \geq 1 \cdot nlog(n)$$

ולכן $nlog^2(n) = \Omega(nlog(n))$. נניח כי $nlog^2(n) = \Theta(nlog(n))$. אז $nlog^2(n) = O(nlog(n))$ ולכן קיימים $n_0, c > 0$ כך שלכל $n \geq n_0$:

$$nlog^2(n) \leq c \cdot nlog(n) \Rightarrow log(n) \leq c$$

לכן גם עבור $n' = \max\{2^c + 1, n_0 + 1\}$ מתקיים

$$log(n') \leq c \Rightarrow 2^c + 1 \leq n' \leq 2^c$$

בסתירה, ולכן $nlog^2(n) = \Omega(nlog(n))$ הוא היחס החזק ביותר בין הפונקציות.

n^2: עבור $n_0 = 1$ ו $c = 4$, מתקיים לכל $n \geq n_0$:

$$log(n) \leq n \Rightarrow log(\sqrt{n}) \leq \sqrt{n}$$

$$log(n) = 2log(\sqrt{n}) \leq 2\sqrt{n}$$

$$\Rightarrow log^2(n) \leq 4n$$

$$\Rightarrow nlog^2(n) \leq 4n^2$$

ולכן $nlog^2(n) = O(n^2)$.

<u>$2^{log^2(n)}$:</u>

נציין כי $2^{log^2(n)} = n^{log(n)}$.

כי

$$2^{log^2(n)} = (2^{log(n)})^{log(n)} = n^{log(n)}$$

n :

נראה $2^{log^2(n)} = n^{log(n)} = \Omega(n)$ **:**

עבור $n_0 = 2$ ו $c = 1$ לכל $n > n_0$:

$$c * n = n \leq n^{log(n)}$$

(כי $1 < log(n)$ עבור $n > 2$)

ולכן מתקיימת הגדרת $n^{log(n)} = \Omega(n)$.

בנוסף,

נראה $n^{log(n)} \neq O(n)$:

נניח בשלילה ש $n^{log(n)} = O(n)$,

לכן קיימים n_0 ו c כך שלכל $n > n_0$:

$$n^{log(n)} \leq cn \Rightarrow n^{log(n)-1} \leq c$$

אבל עבור $n > max\{c, 4, n_0\}$:

$log(n) - 1 > 1$ ו $n > c$ ולכן:

$$c < n \leq n^{log(n)-1} \leq c$$

וזו סתירה.

מכאן שהיחס החזק ביותר המתקיים הוא :

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

n^2:

נראה $n^{log(n)} = \Omega(n^2)$:

עבור $n_0 = 8$ ו $c = 1$ לכל $n > n_0$:

$$cn^2 = n^2 \leq n^3 \underbrace{\leq}_{3<log(n)} \leq n^{log(n)}$$

ולכן מתקיימת הגדרת

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

בנוסף,

נראה $n^{log(n)} \neq O(n^2)$:

נניח בשלילה ש $n^{log(n)} = O(n^2)$,

לכן קיימים n_0 ו c כך שלכל $n > n_0$:

$$n^{log(n)} \leq cn^2 \Rightarrow n^{log(n)-2} \leq c$$

אבל עבור $n > max\{8, c, n_0\}$ מתקיים:

$log(n) - 2 > 1$, $n > c$ ולכן:

$$c < n \leq n^{log(n)-2} \leq c$$

וזו סתירה.

לכן ההיחס החזק ביותר המתקיים הוא

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

$nlog(n)$:

נראה $n^{log(n)} = \Omega(nlog(n))$:

הראנו ש $n^{log(n)} = \Omega(n^2)$,

לכן קיימים c_1וn_1כך ש לכל $n > n_1$:

$$c_1n^2 \leq n^{log(n)}$$

וראינו בהרצאה ש $n^2 = \Omega(nlogn)$,

לכן קיימים c_2וn_2כך ש לכל $n > n_2$:

$c_2n \leq n^2 \Rightarrow c_2c_1nlog(n) \leq c_1n^2 \leq n^{log(n)}$

ולכן עבור $c = c_1 * c_2$ ו $n_0 = max\{n_1, n_2\}$:

$cnlog(n) \leq n^{log(n)}$

ומתקיימת הגדרת

$n^{log(n)} = \Omega(nlog(n))$.

נראה בנוסף ש

$n^{log(n)} \neq O(nlog(n))$

הראנו כבר ש

$n^{log(n)} \neq O(n^2)$

לכן לכל n_0 ולכל $c > 0$ קיים $n > n_0$ כך ש :

$cnlog(n) < cn^2 < n^{log(n)}$

ולכן נשללת הגדרת $n^{log(n)} = O(nlog(n))$.

<u>$log(n!)$</u>:

n : נבחר $n_0 = 4$ ו$c = 1$. נראה ראשית כי לכל $n \geq n_0$, מתקיים $n! \geq 2^n$. עבור $n_0 = 4$ מתקיים

$$n_0! = 1 \cdot 2 \cdot 3 \cdot 4 = 24 > 16 = 2^{n_0}$$

נניח כי עבור n', מתקיים $n'! \geq 2^{n'}$. אז עבור $n = n' + 1$ נקבל:

$$n! = n'! \cdot n \geq 2^{n'} \cdot n \underbrace{\geq}_{n \geq 4} 2^{n'} \cdot 2 = 2^{n'+1} = 2^n$$

ולכן מאינדוקציה לכל $n \geq n_0$ מתקיים

$$n! \geq 1 \cdot 2^n \Rightarrow log(n!) \geq 1 \cdot log(2^n) \Rightarrow log(n!) \geq 1 \cdot n$$

כלומר $log(n!) = \Omega(n)$.

$nlog(n)$: עבור $n_0 = 1, c = 1$ לכל $n \geq n_0$ מתקיים :

$$log(n!) = log(1 \cdot 2 \cdot \ldots \cdot n) = log(1) + log(2) + \ldots + log(n)$$

$log(n)$ היא פונקציה עולה ולכן לכל $1 \leq k \leq n$ מתקיים $log(k) \leq log(n)$. כלומר :

$$log(1) + \ldots + log(n) \leq log(n) + \ldots + log(n) = nlog(n)$$

$$\Rightarrow log(n!) \leq 1 \cdot nlog(n)$$

ולכן $log(n!) = O(nlog(n))$. עבור $c = \frac{1}{4}$ ו$n_0 \geq 4$ לכל $n \geq n_0$ מתקיים:

$$log(n!) = log(1) + \ldots + log\left(\frac{n}{2}\right) + \ldots + logn(n) \geq log\left(\frac{n}{2}\right) + \ldots + log(n)$$

$$\geq log\left(\frac{n}{2}\right) + \ldots + log\left(\frac{n}{2}\right) = \frac{n}{2}log\left(\frac{n}{2}\right) = \frac{n}{2}\left(log(n) - log(2)\right) = \frac{n}{2}\left(log(n) - 1\right)$$

$$= \frac{n}{2}log(n) - \frac{n}{2} = \frac{n}{4}log(n) + \frac{n}{4}log(n) - \frac{n}{4} \cdot 2 = \frac{n}{4}log(n) + \frac{n}{4}\left(log(n) - 2\right) \underbrace{\geq}_{n \geq 4} \frac{1}{4}nlog(n)$$

כפי שנראה בתרגול. ולכן מתקיים גם $log(n!) = \Omega(nlog(n))$ ובסך הכל $log(n!) = \Theta(nlog(n))$.

n^2: עבור $n_0 = 0$ ו$c = 1$ לכל $n \geq n_0$ מתקיים:

$$log(n!) = log(1 \cdot \ldots \cdot n) = \sum_{k=1}^{n} log(k) \leq \sum_{k=1}^{n} k = \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2}$$

לכל n טבעי מתקיים $n \leq n^2$ ולכן $\frac{n}{2} \leq \frac{n^2}{2}$ כלומר:

$$log(n!) \leq \frac{n^2}{2} + \frac{n}{2} \leq \frac{n^2}{2} + \frac{n^2}{2} = n^2$$

ולכן $log(n!) = O(n^2)$.

$n^{log(3)}$:

---

n:

נראה $n^{log(3)} = \Omega(n)$:

עבור $n_0 = 1$ ועבור $c = 1$ מתקיים כי לכל $n > n_0$:

$cn = n \leq n^{log(3)}$

ומתקיימת הגדרת

$n^{log(3)} = \Omega(n)$.

נראה כי $n^{log(3)} \neq O(n)$:

נניח בשלילה ש $n^{log(3)} = O(n)$,

לכן קיימים n_0 ו c כך שלכל $n > n_0$:

$n^{log(3)} \leq cn \Rightarrow n^{log(3)-1} \leq c$

אבל $n^{log(3)-1}$ שואף ל∞ כשה n שואף ל∞ וזו סתירה.

$nlogn$:

נראה $n^{log(3)} = \Omega(nlogn)$**:**

ראינו בתרגול שלכל $\epsilon > 0$ מתקיים:

$logn = o(n^{\epsilon})$

לכן לכל $c > 0$ קיים $n_0 > 0$ כך שלכל $n > n_0$:

$logn \leq cn^{\epsilon}$

לכן עבור $c = 1$ מתקיים:

$nlogn \leq n * n^{log(3)-1} = n^{log(3)}$

ומתקיימת הגדרת

$n^{log(3)} = \Omega(nlogn)$.

נראה כי $n^{log(3)} \neq O(nlogn)$:

נניח בשלילה ש $n^{log(3)} = O(nlogn)$,

לכן קיימים n_0 ו c כך שלכל $n > n_0$:

$n^{log(3)} \leq cnlogn \Rightarrow n^{log(3)-1} \leq clogn$

אבל

$logn = o(n^{\epsilon})$ ולכן

לכל $c > 0$ קיים $n_0 > 0$ כך שלכל $n > n_0$:

$logn \leq cn^{\epsilon}$

ומכך נובע ש

$c \cdot logn < n^{log(3)-1} \leq c \cdot logn$

וזו סתירה.

היחס החזק ביותר הוא $n^{log(3)} = \Omega(nlogn)$.

n^2:

נראה $n^{log(3)} = O(n^2)$**:**

עבור $n_0 = 1$ ועבור $c = 1$ מתקיים כי לכל $n > n_0$:

$n^{log(3)} \leq n^2$ (כי $log3 < 2$)

נראה בנוסף ש $n^{log(3)} \neq \Omega(n^2)$:

נניח בשלילה כי $n^{log(3)} = \Omega(n^2)$,

לכן קיימים n_0 וc כך ש לכל $n > n_0$:

$cn^2 \leq n^{log3} \Rightarrow c \leq n^{log3-2}$

אבל $log3 - 2 < 0$ ולכן n^{log3-2} שואף ל0 וזו סתירה לכך שקיים קבוע חיובי שחוסם את הסדרה מלרע.

לכן היחס החזק ביותר המתאים הוא:

$n^{log(3)} = O(n^2)$

(ב)

נראה הגדרה שקולה:

נראה שלכל $\epsilon > 0$ מתקיים:

$$\lim_{n\to\infty} \frac{loglogn}{log^{\epsilon}(n)} = 0$$

נראה ש

$$\lim_{x\to\infty} \frac{loglogx}{log^{\epsilon}(x)} = 0$$

ולכן הגבול שצריך להוכיח נובע לפי משפט היינה מאינפי 1מ.

$$\lim_{x\to\infty} \frac{loglogx}{log^{\epsilon}(x)} \underbrace{=}_{lopital} \lim_{x\to\infty} \frac{\frac{1}{logx} * \frac{1}{x}}{\epsilon log^{\epsilon-1}(x) * \frac{1}{x}} = \lim_{x\to\infty} \frac{1}{\epsilon log^{\epsilon}(x)} = 0$$

(ג)

יהי $c > 0$. נפצל למקרים.

1. אם $c \geq 1$: נתון $f(n) = \omega(log(n))$ ולכן קיים עבורו $n_0 > 0$ כך ש$f(n) \geq c \cdot log(n)$.

$$f(n) \geq c \cdot log(n) \Rightarrow 2^{f(n)} \geq \left(2^{log(n)}\right)^c$$

$$\Rightarrow 2^{f(n)} \geq n^c \Rightarrow c \cdot 2^{f(n)} \geq c \cdot n^c \geq n$$

$$\Rightarrow n \leq c \cdot 2^{f(n)}$$

ולכן $n = o\left(2^{f(n)}\right)$.

2. אם $0 < c < 1$: נסמן $c_1 = log\left(\frac{1}{c}\right) + 1$. $c < 1$ ולכן $\frac{1}{c} > 1$ כלומר $log\left(\frac{1}{c}\right) > 0$ ו $c_1 > 1$. נתון $f(n) = \omega(log(n))$. לכן קיים $n_0 > 0$ כך שלכל $n \geq n_0$ מתקיים

$$f(n) \geq \left(log\left(\frac{1}{c}\right) + 1\right) log(n)$$

נסמן $n_2 = \max\{n_0, 2\}$ ואז לכל $n \geq n_2$:

$$\Rightarrow 2^{f(n)} \geq 2^{\left(log\left(\frac{1}{c}\right)+1\right)log(n)}$$

$$\Rightarrow 2^{f(n)} \geq \left(2^{log(n)}\right)^{log\left(\frac{1}{c}\right)+1}$$

$$\Rightarrow 2^{f(n)} \geq n^{log\left(\frac{1}{c}\right)+1}$$

$$\Rightarrow c \cdot 2^{f(n)} \geq c \cdot n^{log\left(\frac{1}{c}\right)+1} = c \cdot n^{log\left(\frac{1}{c}\right)} \cdot n \underbrace{\geq}_{n\geq 2} c \cdot 2^{log\left(\frac{1}{c}\right)} \cdot n = c \cdot \frac{1}{c} \cdot n = n$$

$$\Rightarrow n \leq c \cdot 2^{f(n)}$$

ולכן גם במקרה זה $n = o\left(2^{f(n)}\right)$.

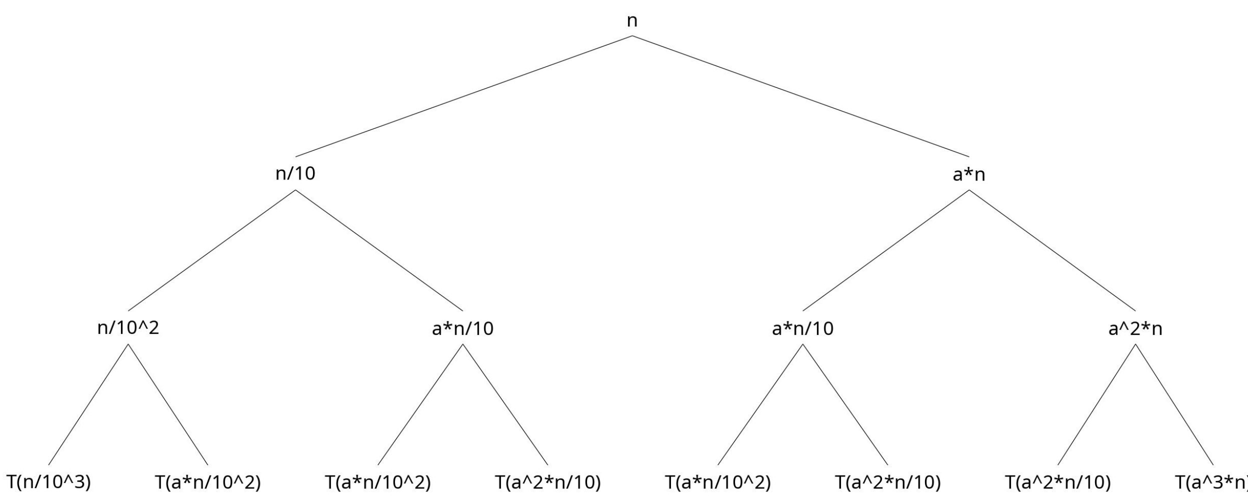
שאלה 3

(א)

נתונה המשוואה הרקורסיבית:

$$T(n) = T\left(\frac{n}{10}\right) + T(a \cdot n) + n$$

נתחיל לפתח לעץ רקורסיה:



נראה באינדוקציה כי סכום הרמה הk הוא $n \cdot \left(a + \frac{1}{10}\right)^k$, כאשר הרמה הראשונה היא $k = 0$.

<u>בסיס:</u> האיבר היחיד ברמה $k = 0$ הוא n ולכן סכומה הוא $n = n \cdot \left(a + \frac{1}{10}\right)^0$.

<u>צעד:</u> נניח כי סכום הרמה הk הוא $n \cdot \left(a + \frac{1}{10}\right)^k$. נסמן כל צומת ברמה ה$k$ כn'_i עבור $1 \leq i \leq 2^k$. עבור כל צומת n'_i, יש בדיוק שני צמתים ברמה ה$k+1$: $a \cdot n'_i, \frac{1}{10} \cdot n'_i$. לכן סכום השורה ה$k+1$ הוא:

$$\sum_{i=1}^{2^k} a \cdot n'_i + \frac{1}{10} n'_i = \sum_{i=1}^{2^k} n'_i \left(a + \frac{1}{10}\right) = \left(a + \frac{1}{10}\right) \sum_{i=1}^{2^k} n'_i$$

אבל $\sum_{i=1}^{2^k} n_i'$ הוא פשוט סכום הצמתים ברמה הk, שלפי הנחת האינדוקציה מקיים $\sum_{i=1}^{2^k} n_i' = n \cdot \left(a + \frac{1}{10}\right)^k$ ולכן סכום השורה ה$k+1$ הוא:

$$\sum_{i=1}^{2^k} a \cdot n_i' + \frac{1}{10} n_i' = \left(a + \frac{1}{10}\right) \sum_{i=1}^{2^k} n_i' = \left(a + \frac{1}{10}\right) \cdot n \cdot \left(a + \frac{1}{10}\right)^k = n \cdot \left(a + \frac{1}{10}\right)^{k+1}$$

כנדרש.

בסך הכל, $T(n)$ שווה לסכום כל הרמות. ניתן למצוא חסם מלעיל ומלרע ל$T(n)$ שכן סכום כל רמה בנפרד הוא חיובי, ומספר הרמות תלוי בגובה העץ. הקריאות הרקורסיביות הן מהצורה $T(a \cdot n)$ ו $T\left(\frac{n}{10}\right)$ ולכן גובה העץ תחום בין $m = \min\left\{log_{10}(n), log_{\frac{1}{a}}(n)\right\}$ ו $M = \max\left\{log_{10}(n), log_{\frac{1}{a}}(n)\right\}$. לכן:

$$\sum_{k=0}^{m} n \cdot \left(a + \frac{1}{10}\right)^k \leq T(n) \leq \sum_{k=0}^{M} n \cdot \left(a + \frac{1}{10}\right)^k$$

$$\Rightarrow n \cdot \sum_{k=0}^{m} \left(a + \frac{1}{10}\right)^k \leq T(n) \leq n \cdot \sum_{k=0}^{M} \left(a + \frac{1}{10}\right)^k$$

נבדוק עבור אילו ערכי a מתקיים $T(n) = \omega(n)$, כלומר $\lim_{n\to\infty} \frac{n}{T(n)} = 0$.

$$\frac{1}{\sum_{k=0}^{m} \left(a + \frac{1}{10}\right)^k} \leq \frac{n}{T(n)} \leq \frac{1}{\sum_{k=0}^{M} \left(a + \frac{1}{10}\right)^k}$$

נשים לב כי כיוון ש - m ו - M הם לוגריתמים בבסיסים שונים של n, מתקיים

$$\lim_{n\to\infty} \sum_{k=0}^{m} \left(a + \frac{1}{10}\right)^k = \lim_{n\to\infty} \sum_{k=0}^{M} \left(a + \frac{1}{10}\right)^k = \sum_{k=0}^{\infty} \left(a + \frac{1}{10}\right)^k$$

ולכן מכלל הסנדוויץ' מתקיים $\lim_{n\to\infty} \frac{n}{T(n)} = \lim_{n\to\infty} \frac{1}{\sum_{k=0}^{m}\left(a+\frac{1}{10}\right)^k}$.

לכן נרצה למצוא מתי מתקיים $\frac{1}{\sum_{k=0}^{\infty}\left(a+\frac{1}{10}\right)^k} = 0$ כלומר מתי מתקיים $\sum_{k=0}^{\infty} \left(a + \frac{1}{10}\right)^k = \infty$.
הטור $\sum_{k=0}^{\infty} \left(a + \frac{1}{10}\right)^k$ הוא טור גיאומטרי ולכן

$$\sum_{k=0}^{\infty} \left(a + \frac{1}{10}\right)^k = \infty \iff a + \frac{1}{10} \geq 1 \iff a \geq \frac{9}{10}$$

לכן $a = \frac{9}{10}$ הוא הערך המינימלי של a עבורו מתקיים $\lim_{n\to\infty} \frac{1}{\sum_{k=0}^{m}\left(a+\frac{1}{10}\right)^k} = 0$ כלומר $\lim_{n\to\infty} \frac{n}{T(n)} = 0$ כלומר $T(n) = \omega(n)$.

כאשר $a=\frac{9}{10}$, מתקיים $\sum_{k=0}^{log(n)}\left(a+\frac{1}{10}\right)^k=\sum_{k=0}^{log(n)}(1)^k=\frac{log(n)(log(n)+1)}{2}$. לכן:

$$n\cdot\frac{\left\lfloor log_{\max\{10,\frac{1}{a}\}}(n)\right\rfloor\left(\left\lfloor log_{\max\{10,\frac{1}{a}\}}\right\rfloor(n)+1\right)}{2}\le n\cdot\sum_{k=0}^{m}n\cdot\left(a+\frac{1}{10}\right)^k\le T(n)$$

$$T(n)\le n\cdot\sum_{k=0}^{M}\left(a+\frac{1}{10}\right)^k\le n\cdot\frac{\lceil log_{\min\{10,\frac{1}{a}\}}(n)\rceil\left(\lceil log_{\min\{10,\frac{1}{a}\}}(n)\rceil(n)+1\right)}{2}$$

$$n\cdot\frac{\left\lfloor log_{\max\{10,\frac{1}{a}\}}(n)\right\rfloor\left(\left\lfloor log_{\max\{10,\frac{1}{a}\}}\right\rfloor(n)+1\right)}{2}\le T(n)\le n\cdot\frac{\lceil log_{\min\{10,\frac{1}{a}\}}(n)\rceil\left(\lceil log_{\min\{10,\frac{1}{a}\}}(n)\rceil+1\right)}{2}$$

ולכן $T(n)=\Theta(n\cdot log^2(n))$

(ב)

1.

$$T(n)=4T(\frac{n}{2})+n^3+nlog^2(n)$$

נראה ש$T(n)=\Theta(n^3)$ לפי הסעיף השלישי של משפט המאסטר:

בהצגה:

$$T(n)=a(T(n/b)+f(n)$$

$$a=4,b=2,f(n)=n^3+nlog^2(n)$$

נראה שמתקיים:

$f(n)=\Omega(n^{log_b a+\epsilon})$ עבור $\epsilon=1$

$$n^{log_b a+\epsilon}=n^{2+1}=n^3$$

ואכן מתקיים כי:

$$n^3+nlog^2(n)=\Omega(n^3)$$

(כי $n^3+nlog^2(n)\ge n^3$ לכל $n>1$).

בנוסף,

$$af(n/b)=4(\frac{n}{2})^3+2nlog^2(n/2)=\frac{n^3}{2}+2n(log(n)-1)^2\le.$$

$$.\le\frac{n^3}{2}+2nlog^2(n)\underbrace{\le}_{*n_0\text{n> for}}\frac{3n^3}{4}\le\frac{3}{4}(n^3+nlog^2n)=\frac{3}{4}f(n)$$

)

נראה*:

נראה שקיים n_0 כך שלכל $n>n_0$

$$2nlog^2(n)\le\frac{n^3}{4}$$

נראה ש $nlog^2(n)=o(n^3)$ ובאופן שקול $\lim_{n\to\infty}\frac{nlog^2(n)}{n^3}=0$:

$$\frac{nlog^2(n)}{n^3} = \frac{log^2(n)}{n^2} \underset{n\to\infty}{\to} 0$$

לכן לכל $c > 0$ ובפרט עבור $c = \frac{1}{8}$ קיים n_0 כך שלכל $n > n_0$:

$$nlog^2(n) \leq cn^3$$

(

מצאנו שתנאי הסעיף השלישי של משפט המאסטר מתקיימים ולכן:

$$T(n) = \Theta(f(n))$$

.2

$$T(n) = log((n^2)!)$$

נסמן $n^2 = m$:

$$T(n) = T(\sqrt{m}) = log(m!) = \Theta(mlogm) = \Theta(n^2log(n^2)) = .$$

$$. = \Theta(2n^2log(n)) = \Theta(n^2logn)$$

.3

$$T(n) = T(\sqrt{n}) + n^2$$

נסמן $n = 2^m$ או לחילופין $m = log(n)$:

ונגדיר פונקציה

$$S(m) = T(n)$$

$$S(m) = T(n) = T(2^m) = T(2^{m/2}) + 2^{2m} = S(m/2) + 4^m$$

נראה $S(m) = \Theta(4^m)$ בעזרת הסעיף השלישי של משפט המאסטר:

$$f(m) = 4^m, a = 1, b = 2$$

נראה ש $f(m) = \Omega(m^{log_b a + \epsilon})$ עבור $\epsilon = 1$:

צ"ל: $4^m = \Omega(m)$

נראה ש $m = o(4^m)$:

$$\lim_{m\to\infty} \frac{m}{4^m} \underbrace{=}_{lopital} \lim_{m\to\infty} \frac{1}{ln(4)4^m} = 0$$

לכן לכל $c > 0$ ובפרט עבור $c = 1$ קיים m_0 כך שלכל $m > m_0$

$$m \leq 4^m$$

ומתקיימת הגדרת $4^m = \Omega(m)$.

נראה את התנאי הנוסף של המשפט:

קיים $1 > c = \frac{1}{2}$ ו $m_0 = 1$ כך שלכל $m > m_0$ מתקיים:

$$a * f(m/b) = 2^m \leq \frac{1}{2} * 4^m = cf(m)$$

תנאי הסעיף השלישי של משפט המאסטר מתקיימים ולכן:

$$S(m) = \Theta(4^m) \Rightarrow \boldsymbol{T(n)} = \Theta(4^{log(n)}) = \Theta(2^{2logn}) = \boldsymbol{\Theta(n^2)}$$

.4

נסמן

$2^m = n$ כך ש $m = log(n)$

ו$S(m) = T(n)$

$$S(m) = T(n) = 16n^4 T(\sqrt{n}) + 2n^8 log^4(n) = ..$$

$$.. = 16 * 16^m S(\frac{m}{2}) + 2 * 16^{2m} * m^4 \Rightarrow ..$$

$$K(m) \underbrace{=}_{\text{K(m) define}} \frac{S(m)}{16^{2m}} = 16 * \frac{S(\frac{m}{2})}{16^m} + 2 * m^4 = 16 * K(\frac{m}{2}) + 2 * m^4$$

נציג את הביטוי כעץ (שרוך):

$$2m^4 \rightarrow 16 * 2 * (\frac{m}{2})^4 \rightarrow 16^2 * 2 * (\frac{m}{4})^4 \rightarrow ... \rightarrow 2 * 16^i * (\frac{m}{2^i})^4$$

עומק העץ הוא :

$$log(m)$$

איבר כללי של צומת בשרוך בדרגה i הוא :

$$2 * 16^i * (\frac{m}{2^i})^4 = 2 * m^4$$

ולכן כדי למצוא ביטוי מפורש לסכום יש לסכום:

$$K(m) = \sum_{i=0}^{log(m)} 2 * m^4 = 2 * m^4 log(m) = \Theta(m^4 log(m) \Rightarrow ..$$

$$K(m) = \frac{S(m)}{16^{2m}} = \Theta(m^4 log(m)) \Rightarrow ..$$

$$T(n) = S(m) = \Theta(16^{2m} m^4 log(m)) = \boldsymbol{\Theta(n^8 log^4(n) log(log(n)))}$$

שאלה 4

מבני נתונים בהם נשתמש עבור פתירת הבעיה:

1) $Days$ -מערך בגודל $D+1$ (כאשר $D = max\{D_i\}$) של פוינטרים לרשימות מקושרות דו כיווניות, שיעקוב אחרי איזה מדינות צריכות להיות בסגר בכל יום.

כל איבר ברשימה המקושרת, יציין אינדקס של מדינה שצריכה להכנס לסגר ביום הזה .

2) $today$ - משתנה ,מספר טבעי,יציין אינדקס של איבר במערך $Days$ המציין איזה יום היום.

כל יום בסוף הפונקציה lock_down_now יקודם באופן הבא : $today = (today + 1)\%(D+1)$.

3) $countries$- מערך המדינות.

מערך בגודל N ולכל איבר בו שלוש שדות:

($freq$) איבר במקום הi במערך יציג D_{i+1} .

($pointer$) פוינטר לאיבר ברשימות המקושרות הנמצאות ב$Days$ המציינות את המדינה ה $i+1$.

($nextLock$) אנדקס המציין את היום ב$Days$ בו כרגע מתוכנן להיות הסגר הבא.

נציג תיאור של מימוש הפונקציות: $- Init(N,D)$

נאתחל מערך פוינטרים $Days$ שגודלו $D+1$. (בגישה לאיבר במערך שעוד לא אותחל יוחזר $null$).

נאתחל מערך $countries$ שגודלו N.

לאחר האתחול המערך $countries$ יהיה ריק.

נאתחל את המשתנה $today$ להיות 0.

אתחול המערכים והמשתנה נעשה ב$O(1)$ כפי שנלמד בהרצאה.

$- Insert(i, D_i, days)$

נבדוק קלט- עבור קלט לא תקין ($i \leq 0$ או $i > N$, $D_i > D$, $days > D$, i already assigned) , נחזיר שגיאה.

נאתחל את האיבר ה$i-1$ במערך $countries$ כך שיחיל את D_i בשדה הראשון.

נחשב $x = (today + days)\%(D+1)$, ניגש ל $Days[x]$ - ונכניס לראש הרשימה המקושרת אותה הוא מחזיק את הערך i.

נקח את הפוינטר לאיבר שהכנסנו לרשימה ונכניס אותו לשדה השני של $countries[i-1]$.

- Precede_lockdown(i,days)

נחשב $x = (countries.nextLock - days)\%(D+1)$,

ניגש ל $countries.pointer$ ונסיר את האיבר המצוין במצביע מהרשימה המקושרת. (הסרה מרשימה מקושרת - $O(1)$)

נכניס לראש הרשימה המקושרת ב$Days[x]$ איבר n בעל האנדקס i.($O(1)$)

לבסוף נעדכן $countries.nextLock = x, countries.pointer = *n$.

כך שבסוף הפונקציה, הסרנו את התכנון הקודם לסגר והוספנו את התכנון החדש המוקדם במספר הימים הרצוי.

שדות $countries$ עודכנו לערכים חדשים הציינים את הסגר הבא .

כל הפעולות המבוצעות הן ב$O(1)$.

- lock_down_now()

ניגש ל$Days[today]$, ועבור כל איבר ברשימה המקושרת שעליו מצביע נבצע הפעולות הבאות:

* נדפיס את האנדקס המופיע באיבר- נסמנו i

* נסיר את האיבר מהרשימה המקושרת

* נקדם את הסגר הבא להיות בעוד D_i ימים, על ידי הוספת איבר x בעל הערך i לרשימה המקושרת ב $Days[(today+D_i)\%(D+1)]$.

*נעדכן את שדות $countries[i-1]$:

$countries[i-1].pointer = *x$

$countries[i-1].nextLock = (today + D_i)\%(D+1)$

סה"כ אנו עוברים על ורק על המדינות שהיו בסגר באותו יום.

לבסוף, לאחר אטרציה על כל האברים ברשימה ,

נקדם $today = (today+1)\%(D+1)$ ובכך נעבור ליום הבא.

האטרטציה על המדינות שצריכות להיות מודפסות היא $O(k)$ כאשר שאר הפעולות הן $O(1)$.

סה"כ סיבוכיות מקום למבנה בכל רגע נתון ברשימות המקושרות שD מחזיקה יש עד N איברים סה"כ -$O(N)$.

המערך $Days$ שגודלו $D+1$ -$O(D)$

המערך $coutnries$ שגודלו N - $O(N)$

סה"כ $O(D+N)$.

שאלה 5

על מנת שנוכל להדפיס את המדינות בסדר הצבעות יורד לפי הסקר בלי מעבר על איברים מיותרים, נשמור את המדינות ברשימה של דירוגים, שלא תצטרך להיות מאותחלת באתחול של המבנה ולכן לא תפגע בסיבוכיות הזמן הנדרש -
כל איבר בדירוג יווצר רק כאשר תתווסף הצבעה שמשנה את מאזן הדירוגים בין המדינות. במקרה הגרוע ביותר לכל מדינה יהיה מספר הצבעות שונה, כלומר איבר משלה ברשימת הדירוגים, ולכן סיבוכיות המקום של הרשימה היא עדיין $O(N)$.
בנוסף, על מנת להדפיס את כל המדינות שלא רוצים לטוס אליהן בלי לעבור על איברים מיותרים, נשמור מערך נפרד של כל המדינות, וכאשר מדינה מקבלת הצבעה, נקשר בין המדינות שסובבות אותה, כך שיהיה ניתן לדלג עליה במעבר על המדינות.

מבנה הנתונים יכלול את השדות הבאים: מערך מספרים שלמים departures באורך N. התא departures[i] שומר את מספר האנשים שהצביעו שהם רוצים לטוס מהמדינה ה-i.
רשימה דו כיוונית ranking כך שכל איבר ברשימה מחזיק רשימת מדינות בעלות אותו מספר הצבעות של אנשים שרוצים לטוס אליהן.
פוינטר top שמצביע לדירוג הגבוה ביותר.
מערך arrivlas לפוינטרים של מדינות, כאשר התא הi במערך תמיד יכיל פוינטר למדינה הi.
מערך avoided של פוינטרים למדינות שאף אחד לא רוצה לטוס אליהן.
מספר first_avoided ששומר את מספר המדינה הראשונה שתודפס על ידי ()Avoided. מאותחל להיות 0.
מספר last_avoided ששומר את מספר המדינה האחרונה שתודפס על ידי ()Avoided. מאותחל להיות N-1.

ואת הפעולות: 1. Init:
מאתחלת את המערכים departures,arrivals וavoided ב$O(1)$ כפי שנלמד בהרצאה.
בשימוש הראשון בכל פוינטר ב avoided וarrivals , הוא יאותחל לטיפוס מדינה המכיל את מספר המדינה, את כמות האנשים שרוצים לטוס אליה, ושני פוינטרים על מנת ליצור רשימה דו כיוונית של מדינות בעלות אותו מספר אנשים שרוצים לטוס אליהן.
2.Fly(j,i):
ניגשת ראשית לdepartures[j] ומגדילה את ערכו ב1. לאחר מכן ניגשת למדינה בarrivals[i]. אם זו גישה ראשונה למדינה זו, היא תאותחל כמדינה עם המספר i. מספר ההצבעות יוגדל ב-1.
כעת נטפל בדירוג המדינה. נסתכל על הדירוג המיידי הגבוה יותר, אם קיים.
אם מספר ההצבעות של המדינות בדירוג הבא שווה למספר ההצבעות החדש של המדינה לאחר שהועלה, נסיר את המדינה מרשימת המדינות בדירוגה הנוכחי. זה יהיה אפשרי ב$O(1)$ כיוון שנשתמש ברשימה דו כיוונית. לאחר מכן נכניס אותה לראש הרשימה של הדירוג הבא, גם זה ב$O(1)$. אם המדינה הייתה לבד בדירוג שלה נשחרר את הדירוג.
אם מספר ההצבעות של המדינות בדירוג הבא גדול יותר או שאין דירוג יותר גבוה:
אם המדינה היא היחידה בדירוג שלה, לא נעשה כלום. הדירוג כעת ייצג את מספר ההצבעות החדש של המדינה.
אם היו מדינות נוספות בדירוג, ניצור דירוג חדש שייצג את מספר ההצבעות החדש של המדינה. נחבר אותו לדירוג הישן של המדינה בתור הדירוג הבא בסידור. אם הדירוג הישן היה הגבוה ביותר, נעביר את הפוינטר top להצביע על הדירוג החדש. אם היה דירוג גבוה יותר אבל לא היה ניתן להכניס את המדינה לתוכו, נחבר אותו לדירוג שיצרנו בתור הדירוג הבא בתור.
3.Arrivals(i):
ניגשת לתא ה i במערך arrivals ומחזירה את מספר ההצבעות של המדינה אליה הוא מצביע. זו פשוט גישה למערך לפי אינדקס ולכן סיבוכיות הזמן היא $O(1)$.

.4 Depatures(j):

מחזירה את ערך התא ה j במערך depatures. זו פשוט גישה למערך לפי אינדקס ולכן סיבוכיות הזמן היא $O(1)$.

.5 Favored(k):

הפויינטר top כאמור תמיד מצביע לדירוג הגבוה ביותר. נעבור על רשימת המדינות בדירוג הזה, ונתחיל להדפיס אותן, עד שנדפיס k מדינות. עבור כל דירוג שהדפסנו את כל המדינות בו לפני שסיימנו להדפיס k מדינות, נעבור לדירוג הנמוך יותר אחריו ונמשיך.

.6 Avoided():

האיטרציה על המערך avoided מתבססת על העיקרון הבא: בכל פעם שמדינה מקבלת הצבעה ולא תהיה מודפסת על ידי Avoided() יותר, המדינה שלפניה והמדינה שאחריה בסדר הריצה על avoided יאותחלו במערך avoided אם הן לא היו מאותחלות עד עכשיו, ויחוברו ביניהן לרשימה דו כיוונית. אם הן לא היו מאותחלות, אז הן ודאי באינדקס i-1 או i+1 בהתאמה, ולכן ניתן לאתחל את מספר המדינה של כל אחת מהן.

כאשר נרוץ על המערך avoided נעזר במשתנה counter. בכל איטרציה של הפונקציה, החל מהאיבר הראשון, counter יודפס. אם האיבר במיקום counter לא מאותחל, או שאינו מחובר לאף מדינה אחרת, counter יקודם באחד ויודפס שוב.

אחרת, סימן שקיים רצף מדינות שקיבלו הצבעות החל מהאיבר הבא במערך, ולכן במקום לעבור לאיבר הבא בavoided, נעבור לאיבר הבא ברשימה המחוברת לאיבר הנוכחי.

הוא יהיה פוינטר למדינה הבאה שלא קיבלה הצבעות, ובהכרח יהיה מאותחל. לכן נוכל לשנות את counter לערך המדינה ששמור באיבר עצמו, להדפיס את counter, והתהליך חוזר חלילה, עד שcounter מגיע לערך last_avoided.

אם בפונקציה Fly(j,i) מתווספת הצבעה למדינה שעד עכשיו הייתה הראשונה בסדר הקריאה של Avoided(), first_avoided יעבור לשמור את מספר האיבר הבא אחריו בסדר ההדפסה-

כלומר, מספר המדינה אליה מצביע מלפנים הפויינטר שעד עכשיו היה האיבר הראשון בסדר ההדפסה אם הייתה מחוברת אליו מדינה, וfirst_avoided+1 אם לא.

אם בפונקציה Fly(j,i) מתווספת הצבעה למדינה שעד עכשיו הייתה האחרונה בסדר הקריאה של Avoided(), last_avoided יעבור לשמור את מספר האיבר שלפניו בסדר ההדפסה-

כלומר, מספר המדינה אליה מצביע מאחור הפויינטר שעד עכשיו היה האיבר האחרון בסדר ההדפסה אם הייתה מחוברת אליו מדינה, וlast_avoided-1 אם לא.