מבני נתונים - תרגיל בית 1

20 בנובמבר 2020

שאלה 1

סעיף 1 - הטענה לא נכונה

ניקח לדוגמא $f_1(n)=\frac{1}{n}$. נבחר $c=1$ ונבחר $n_0=1$ ואז לכל $n>n_0$ מתקיים $\frac{1}{n}\le 1$ ולכן $f_1(n)=O(1)$.

נניח בשלילה כי $f_1(n)=\Theta(1)$. בפרט מההרצאה נסיק כי $f_1(n)=\Omega(1)$.

כלומר קיימים קבועים c, n_0' כך שלכל $n>n_0$ מתקיים: $c\le\frac{1}{n}$ כלומר $n\le\frac{1}{c}$. בפרט אם נבחר $n=\max\left\{n_0,\lceil\frac{1}{c}\rceil+1\right\}$ נקבל סתירה.

סעיף 2 - הטענה נכונה

$\Leftarrow$: נתון ש-$f_1(n)\le c$ לכל $n>\tilde{n_0}$ עבור c קבוע כלשהו. צ"ל: קיימים d,n_0 כך ש-$d\cdot f_1(n)\le\sqrt{f_1(n)}$ לכל $n>n_0$.

מספיק להראות ש $\sqrt{f_1(n)}\le\frac{1}{d}$ כלומר $f_1(n)\le\frac{1}{d^2}$ (כאשר במעבר האחרון חילקנו ב-$\sqrt{f_1(n)}$ מכיוון שהפונקציה חיובית מהנתון).

נבחר $d=\frac{1}{\sqrt{c}}$ ו-$n_0=\tilde{n_0}$ ומהנתון אנו יודעים כי מתקיים $f_1(n)\le c$ לכל $n>\tilde{n_0}$, כנדרש.

$\Rightarrow$: נתון ש-$\sqrt{f_1(n)}=\Omega(f_1(n))$, כלומר קיימים $\tilde{c},\tilde{n_0}$ כך ש-$\tilde{c}\cdot f_1(n)\le\sqrt{f_1(n)}$. כלומר לאחר חלוקה ב-$\sqrt{f_1(n)}$ נקבל ש-$\sqrt{f_1(n)}\le\frac{1}{\tilde{c}}$ ולאחר העלאה בריבוע נקבל כי $f_1(n)\le\frac{1}{\tilde{c}^2}$. נבחר את $c=\frac{1}{\tilde{c}^2}$ ו-$n_0=\tilde{n_0}$ ואז לכל $n>n_0$ מתקיים $f_1(n)\le\frac{1}{\tilde{c}^2}$ ולכן $f_1(n)=O(1)$, כנדרש.

סעיף 3 - הטענה לא נכונה

נבחר $f_1(n)=n^2$ ובנוסף $f_2(n)=n$ ובנוסף $g_1(n)=g_2(n)=n^3$. נשים לב כי מתקיים $f_1(n)=n^2\le n^3=g_1(n)$ ולכן $f_1(n)=O(g_1(n))$ עבור $c=1$ ו-$n_0=1$.

באופן דומה, מתקיים $f_2(n)=n\le n^3=g_2(n)$ ולכן $f_2(n)=O(g_2(n))$ עבור $c=1$ ו-$n_0=1$.

מתקיים: $\frac{f_1(n)}{f_2(n)}=\frac{n^2}{n}=n$ ובנוסף $\frac{g_1(n)}{g_2(n)}=\frac{n^3}{n^3}=1$ אבל ראינו בהרצאה ש-n איננו $O(1)$. נניח בשלילה כי $n=O(1)$. לכן קיימים c,n_0 כך שלכל $n>n_0$ מתקיים $n\le c$. אך עבור $n=\max\left\{n_0,\lceil c\rceil+1\right\}$ ואז נקבל סתירה לאי שיוויון.

סעיף 4 - הטענה לא נכונה

נבחר $f_1\left(n\right)=n^2$ ו-$g_1\left(n\right)=n$. אכן מתקיים $\left|\log\frac{n^2}{n}\right|=\left|\log n\right|=O\left(\log\left(n\right)\right)$ כי עבור $n_0=1$ ו-$c=1$ מתקיים שלכל $n>1$: $\left|\log n\right|=\log n\leq1\cdot\log n$.

אבל אנו יודעים כי n^2 איננו $O\left(n\right)$ כי אם נניח בשלילה ש-$n^2=O\left(n\right)$ אזי קיימים n_0,c כך שלכל $n>n_0$ מתקיים $n^2\leq c\cdot n$ כלומר $n\leq c$. נבחר $n=\lceil c\rceil+1$ ונקבל סתירה לאי השיוויון.

סעיף 5 - הטענה נכונה

מהנתון נסיק כי :

1. קיימים c_1,n_1 כך שלכל $n>n_1$ מתקיים : $f_1\left(n\right)\leq c_1\cdot g_1\left(n\right)$.
2. קיימים d_1,n_2 כך שלכל $n>n_2$ מתקיים : $f_1\left(n\right)\geq d_1\cdot g_1\left(n\right)$.
3. קיימים c_2,n_3 כך שלכל $n>n_3$ מתקיים : $f_2\left(n\right)\leq c_2\cdot g_2\left(n\right)$, כלומר $\frac{1}{f_2(n)}\leq\frac{1}{c_2\cdot g_2(n)}$.
4. קיימים d_2,n_4 כך שלכל $n>n_4$ מתקיים : $f_2\left(n\right)\geq d_2\cdot g_2\left(n\right)$, כלומר $\frac{1}{f_2(n)}\geq\frac{1}{d_2\cdot g_2(n)}$.

צ"ל: $\frac{f_1(n)}{f_2(n)}=O\left(\frac{g_1(n)}{g_2(n)}\right)$ ובנוסף $\frac{f_1(n)}{f_2(n)}=\Omega\left(\frac{g_1(n)}{g_2(n)}\right)$.

כלומר, עלינו למצוא k,n_0 כך שלכל $n>n_0$ מתקיים $\frac{f_1(n)}{f_2(n)}\leq k\cdot\frac{g_1(n)}{g_2(n)}$.

נסמן $n_0=\max\left\{n_1,n_4\right\}$ ואז לכל $n>n_0$ מתקיים : $\frac{f_1(n)}{f_2(n)}\leq\frac{c_1\cdot g_1(n)}{d_2\cdot g_2(n)}$ (מסעיפים 1+4) , לכן נבחר $k=\frac{c_1}{d_2}$ ולכן ניתן להסיק ש-$\frac{f_1(n)}{f_2(n)}=O\left(\frac{g_1(n)}{g_2(n)}\right)$.

כלומר, עלינו למצוא k',n_0' כך שלכל $n>n_0'$ מתקיים $\frac{f_1(n)}{f_2(n)}\geq k'\cdot\frac{g_1(n)}{g_2(n)}$.

נסמן $n_0'=\max\left\{n_2,n_3\right\}$ ואז לכל $n>n_0'$ מתקיים : $\frac{f_1(n)}{f_2(n)}\geq\frac{c_2\cdot g_1(n)}{d_1\cdot g_2(n)}$(מסעיפים 2+3) , לכן נבחר $k=\frac{c_2}{d_1}$ ולכן ניתן להסיק ש-$\frac{f_1(n)}{f_2(n)}=\Omega\left(\frac{g_1(n)}{g_2(n)}\right)$.

סעיף 6 - הטענה נכונה

נתון כי $\left|\log\left(\frac{f_1(n)}{g_1(n)}\right)\right|=O\left(f_2\left(n\right)\right)$, לכן קיימים n_1,c_1 כך שלכל $n>n_1$ מתקיים : $\left|\log\left(\frac{f_1(n)}{g_1(n)}\right)\right|\leq c_1\cdot f_2\left(n\right)$.

בנוסף, נתון כי $\left|\log\left(\frac{g_1(n)}{g_2(n)}\right)\right|=O\left(f_2\left(n\right)\right)$, לכן קיימים n_2,c_2 כך שלכל $n>n_2$ מתקיים : $\left|\log\left(\frac{g_1(n)}{g_2(n)}\right)\right|\leq c_2\cdot f_2\left(n\right)$.

צ"ל שקיימים c,n_o כך שלכל $n>n_0$ מתקיים $\left|\log\left(\frac{f_1(n)}{g_2(n)}\right)\right|\leq c\cdot f_2\left(n\right)$.

מספיק להוכיח כי $\left|\log f_1\left(n\right)-\log g_2\left(n\right)\right|\leq c\cdot f_2\left(n\right)$.

נבחר $n_0 = \max\{n_1, n_2\}$ ואז לכל $n > n_0$ מתקיים:

$$\begin{aligned}
\left|\log f_1(n) - \log g_2(n)\right| &= \left|\log f_1(n) - \log g_1(n) + \log g_1(n) - \log g_2(n)\right| \\
&\le \left|\log f_1(n) - \log g_1(n)\right| + \left|\log g_1(n) - \log g_2(n)\right| \\
&= \left|\log\left(\frac{f_1(n)}{g_1(n)}\right)\right| + \left|\log\left(\frac{g_1(n)}{g_2(n)}\right)\right| \\
&\le c_1 \cdot f_2(n) + c_2 \cdot f_2(n) \\
&= (c_1 + c_2) \cdot f_2(n)
\end{aligned}$$

לכן ניקח $c = c_1 + c_2$ ואז נקבל כי $\left|\log\left(\frac{f_1(n)}{g_2(n)}\right)\right| = O(f_2(n))$, כנדרש.

שאלה 2

סעיף 1

$g(n)$ \ $f(n)$	n	$n\log n$	n^2
$n\log^2 n$	$_{(1)}\Omega$	$_{(5)}\Omega$	$_{(9)}O$
$2^{\log^2 n}$	$_{(2)}\Omega$	$_{(6)}\Omega$	$_{(10)}\Omega$
$\log n!$	$_{(3)}\Omega$	$_{(7)}\Theta$	$_{(11)}O$
$n^{\log 3}$	$_{(4)}\Omega$	$_{(8)}\Omega$	$_{(12)}O$

1. נבחר $c = 1$ ו-$n_0 = 1$ ואז לכל $n > n_0$ מתקיים $n\log^2 n \ge 1 \cdot n$ מכיוון ש-$\log^2 n \ge 1$ לכל $n \ge 2$.
2. נבחר $c = 1$ ו-$n_0 = 1$ ואז לכל $n > n_0$ מתקיים $2^{\log^2 n} = \left(2^{\log n}\right)^{\log n} = n^{\log n} \ge 1 \cdot n$ אמ"ם $n^{\log(n)-1} \ge 1$ אבל אנו יודעים שמתקיים $n^{\log(n)-1} \ge 1$ מכיוון ש-$\log(n) - 1 \ge 0$ לכל $n \ge 2$ ומכיוון שאנו יודעים ש- n בחזקה אי שלילית יהיה גדול שווה מ-1, קיבלנו את הדרוש.
3. הוכחנו בתרגול 1 שקף 6 כי $\log n! = \Omega(n\log n)$. לכן קיימים $\tilde{n_0}, \tilde{c} > 0$ כך שלכל $n > \tilde{n_0}$ מתקיים $\log n! \ge \tilde{c} \cdot n\log n$. בנוסף, אנו יודעים כי $\tilde{c} \cdot n\log n \ge \tilde{c} \cdot n$ מכיוון ש-$\log n \ge 1$ לכל $n \ge 2$ (לאחר הכפלת אי השיוויון ב-$\tilde{c} \cdot n$). נבחר $n_0 = \max\{2, \tilde{n_0}\}$ ואז לכל $n > n_0$ מתקיים $\log n! \ge \tilde{c} \cdot n\log n \ge \tilde{c} \cdot n$. ולכן סה"כ קיבלנו כי עבור $c = \tilde{c}$ מתקיים $\log n! \ge c \cdot n$, ולכן נסיק כי $\log n! = \Omega(n)$, כנדרש.
4. נבחר $c = 1$ ו-$n_0 = 1$ ואז לכל $n > n_0$ מתקיים $n^{\log 3} \ge 1 \cdot n$ אמ"ם $n^{\log(3)-1} \ge 1$ מכיוון ש-$\log(3) - 1 > 0$ ומכיוון שאנו יודעים ש- n בחזקה חיובית יהיה גדול מ-1, קיבלנו את הדרוש.
5. נבחר $c = 1$ ו-$n_0 = 1$ ואז לכל $n > n_0$ מתקיים $n\log^2 n \ge 1 \cdot n\log n$ מכיוון ש-$\log n \ge 1$ לכל $n \ge 2$.

6. צ"ל שקיימים קבועים $c>0$, n_0 כך שלכל $n>n_0$ מתקיים $n^{\log n}\geq c\cdot n\log n$ אמ"ם $n^{\log(n)-1}\geq c\cdot\log n$. אנו יודעים ש-$\log(n)-1\geq 1$ לכל $n\geq 4$ ולכן $n^{\log(n)-1}\geq n^1=n$. אבל אנו יודעים כי $n=\Omega(\log n)$ מכיוון שעבור $n_1=1$ ו-$c'=1$ מתקיים $n\geq 1\cdot\log n$ לכל $n>n_1$. לכן, אם נסמן $c=c'=1$ ו-$n_0=4$ לכל $n>n_0$ מתקיים: $n^{\log(n)-1}\geq n\geq 1\cdot\log n$, כלומר $n^{\log n}=\Omega(n\log n)$.

7. ראינו בתרגול 1 (שקף 5-6) כי מתקיים $\log n!=\Theta(n\log n)$.

8. צ"ל שקיימים קבועים $c>0$, n_0 כך שלכל $n>n_0$ מתקיים $n^{\log 3}\geq c\cdot n\log n$. מספיק להראות שקיימים קבועים כך ש-$\frac{n^{\log(3)-1}}{c}\geq\log n$. הוכחנו בתרגול 1 שקופית 16 כי לכל $\varepsilon>0$ מתקיים $\log(n)=o(n^{\varepsilon})$. לכן עבור $\varepsilon=\log(3)-1$ נקבל שלכל c' (בפרט עבור $c'=1$) קיים n_1 כך שלכל $n>n_1$ מתקיים $\log(n)\leq 1\cdot n^{\log(3)-1}$. לכן אם נבחר $c=\frac{1}{c'}=1$ נקבל את הדרוש, ולכן נסיק כי $n^{\log 3}=O(n\log n)$.

9. נבחר $c=1$ ו-$n_0=1$ ואז לכל $n>n_0$ מתקיים $n\log^2 n\leq n^2$ מכיוון ש-$n\geq\log^2 n$ לכל $n\geq 2$.

10. נבחר $c=1$ ו-$n_0=3$ ואז לכל $n>n_0$ מתקיים $2^{\log^2 n}=\left(2^{\log n}\right)^{\log n}=n^{\log n}=1\cdot n^2$ אמ"ם $n^{\log(n)-2}\geq 1$ אבל אנו יודעים שמתקיים $n^{\log(n)-2}\geq 1$ מכיוון ש-$\log(n)-2\geq 0$ לכל $n\geq 4$ ומכיוון שאנו יודעים ש- n בחזקה אי שלילית יהיה גדול מ-1, קיבלנו את הדרוש.

11. הוכחנו בתרגול 1 שקף כי $\log n!=O(n\log n)$. לכן קיימים $c>0$, n_0 כך שלכל $n>n_0$ מתקיים $\log n!\leq c\cdot n\log n$. בנוסף אנו יודעים ש-$c\cdot\log n\leq c\cdot n$ לכל $n\geq 2$ ולאחר לאחר הכפלת אי השיוויון ב-n נקבל כי $c\cdot n\log n\leq cn^2$. נבחר $\tilde{n_0}=\max\{2,n_0\}$ ואז לכל $n>\tilde{n_0}$ מתקיים $\log n!\leq c\cdot n\log n\leq cn^2$ כלומר $\log n!\leq c\cdot n^2$ ולכן עבור $\tilde{c}=c$ נקבל כי $\log n!=O(n^2)$.

12. נבחר $c=1$ ו-$n_0=1$ ואז לכל $n>n_0$ מתקיים $n^{\log 3}\leq 1\cdot n^2$ אמ"ם $n^{\log(3)-2}\leq 1$ מכיוון ש-$\log(3)-2<0$ ומכיוון שאנו יודעים ש- n בחזקה שלילת יהיה קטן מ-1 לכל $n>1$, קיבלנו את הדרוש.

סעיף 2

יהי $\varepsilon>0$. צ"ל שלכל $c>0$ קיים n_0 כך שלכל $n>n_0$ מתקיים: $\log\log n\leq c\cdot\log^{\varepsilon}n$, או באופן שקול כפי שלמדנו בתרגול 1 צ"ל ש-$\lim_{n\to\infty}\frac{\log\log n}{\log^{\varepsilon}n}=0$.
נשתמש בכלל לופיטל. מתקיים:

$$\lim_{n\to\infty}\frac{\log\log n}{\log^{\varepsilon}n}=\lim_{n\to\infty}\frac{\frac{1}{\log(n)\cdot\ln(2)}\cdot\frac{1}{\ln(2)\cdot n}}{\varepsilon\cdot\log^{\varepsilon-1}(n)\cdot\frac{1}{\ln(2)\cdot n}}$$
$$=\lim_{n\to\infty}\frac{1}{\varepsilon\cdot\ln(2)\cdot\log^{\varepsilon}(n)}$$
$$\overset{(*)}{=}0$$

הסבר ל-$(*)$ - מאינפי 1 אנו יודעים כי לכל $\varepsilon > 0$ מתקיים $\log^{\varepsilon}(n) \xrightarrow{n\to\infty} \infty$ ולכן נסיק כי $\frac{1}{\varepsilon\cdot\ln(2)\cdot\log^{\varepsilon}(n)} \xrightarrow{n\to\infty} 0$, כנדרש.

סעיף 3

מהנתון לכל $c > 0$ קיים n_0 כך שלכל $n > n_0$ מתקיים $f(n) \geq c\cdot\log(n)$ ולכן לאחר העלאה מעריכית נקבל ש-$2^{f(n)} \geq 2^{c\cdot\log n} = \left(2^{\log n}\right)^c = n^c$.

צ"ל ש-$n = o\left(2^{f(n)}\right)$ - כלומר שלכל $d > 0$ קיים $n_1 > 0$ כך שלכל $n > n_1$ מתקיים $n \leq d\cdot 2^{f(n)}$.

נניח בשלילה שקיים $d > 0$ כך שלכל $n > 0$ קיים $n_1 > n$ כך ש-$n_1 \geq d\cdot 2^{f(n_1)}$. נבחר $c = \log_{n_1}\left(\frac{1}{d}\right) + 2$ (הסבר בהמשך) ונבחר $n = n_0$ (n_0 ששייך ל-c שבחרנו זה עתה) ואז קיים מהנחת שלילה $n_1 > n_0$ כך ש-$n_1 \geq d\cdot 2^{f(n_1)}$. בנוסף, מהנתון לכל $n > n_0$ - בפרט עבור n_1 - (מכיוון ש-$n_1 > n_0$) אז מתקיים ש-$2^{f(n_1)} \geq (n_1)^c$. כלומר, סה"כ קיבלנו מטרנזיטיביות ש-$n_1 \geq d\cdot(n_1)^c$, כלומר $\frac{1}{d} \geq (n_1)^{c-1}$ אמ"ם $\log_{n_1}\left(\frac{1}{d}\right) + 1 \geq c$ (לאחר הפעלת $\log_{n_1}$). כלומר קיבלנו $\log_{n_1}\left(\frac{1}{d}\right) + 1 \geq \log_{n_1}\left(\frac{1}{d}\right) + 2$, כלומר $1 \geq 2$ וזו סתירה.

שאלה 3

סעיף א'

ראשית, נפתח את הביטוי:

$$\begin{aligned} T\left(n\right) &= T\left(\frac{n}{10}\right)+T\left(a\cdot n\right)+n \\ &= T\left(\frac{n}{10^{2}}\right)+T\left(a\cdot\frac{n}{10}\right)+\frac{n}{10}+T\left(a\cdot\frac{n}{10}\right)+T\left(a^{2}\cdot n\right)+a\cdot n+n \\ &= T\left(\frac{n}{10^{3}}\right)+T\left(a\cdot\frac{n}{10^{2}}\right)+\frac{n}{10^{2}}+2\cdot T\left(a\cdot\frac{n}{10^{2}}\right)+2T\left(a^{2}\cdot\frac{n}{10}\right) \\ &\quad +2a\cdot\frac{n}{10}+T\left(a^{2}\cdot\frac{n}{10}\right)+T\left(a^{3}\cdot n\right)+a^{2}n+\left(n+\frac{n}{10}\right)+an \end{aligned}$$

נשים לב שסכום כל שורה בעץ מסתכמת ל-$n\left(\frac{1}{10}+a\right)^{h}$ כאשר h הוא עומק העץ.

כלומר - $T\left(n\right)=\sum_{k=0}^{h}n\left(\frac{1}{10}+a\right)^{k}=n\cdot\sum_{k=0}^{h}\left(\frac{1}{10}+a\right)^{k}$.

נסמן $q=\frac{1}{10}+a$. אנו יודעים כי הטור $\sum_{k=0}^{n}\left(q\right)^{k}$ מתכנס לקבוע אמ"ם $|q|<1$.

נניח כי הטור $\sum_{k=0}^{n}\left(q\right)^{k}$ מתכנס ל-c_1 כאשר נשאיף את n לאינסוף. אזי בפרט עבור $|q|<1$ מתקיים $\frac{c_1}{2}\cdot n\leq T\left(n\right)\leq c_1\cdot n$

כלומר $T\left(n\right)=\Theta\left(n\right)$ אבל אז נקבל כי $T\left(n\right)\neq\omega\left(n\right)$ כי מתקיים $\frac{T(n)}{n}\leq c_1$ כלומר $\lim\limits_{n\to\infty}\frac{T(n)}{n}\neq\infty$.

לכן בהכרח $|q|\geq1$. כלומר $\frac{1}{10}+a\geq1$. דהיינו $a\geq\frac{9}{10}$. כעת נבדוק האם עבור $a=\frac{9}{10}$ מתקיים $T\left(n\right)=\omega\left(n\right)$.

נשים לב כי העומק מצד ימין הוא $\log_{\frac{1}{a}}\left(n\right)$, והעומק מצד שמאל הוא $\log_{10}\left(n\right)$.

עבור $a=\frac{9}{10}$ נקבל ש-$\log_{\frac{10}{9}}\left(n\right)\geq\log_{10}\left(n\right)$ ולכן העומק המינימלי הוא $\log_{10}\left(n\right)$. לכן $n\cdot\sum_{k=0}^{\lfloor\log_{10}(n)\rfloor}\left(\frac{1}{10}+a\right)^{k}\leq T\left(n\right)$

יהי $c>0$. נבחר $n_0=\lceil10^{c+1}\rceil$ ואז לכל $n>n_0$ מתקיים:

$$T\left(n\right)\geq n\cdot\lfloor\log_{10}\left(n\right)\rfloor\geq n\left(\log_{10}\left(n\right)-1\right)\overset{(n\geq\lceil10^{c+1}\rceil)}{\geq}n\cdot\log_{10}\left(\lceil10^{c+1}\rceil\right)-n\geq n\cdot\log_{10}\left(10^{c+1}\right)-n=n\cdot\left(c+1\right)-n=c\cdot n$$

סה"כ, קיבלנו כי מתקיים $T\left(n\right)=\omega\left(n\right)$, כנדרש.

קיבלנו כי ה-a המינימלי המקיים $T\left(n\right)=\omega\left(n\right)$ הוא $\frac{9}{10}$.

סעיף ב'

סעיף ב1

ניקח איברים מהצורה $n=2^{k}$ ולאחר מכן נוכיח עבור n כללי.

מתקיים:

$$
\begin{aligned}
T\left(n\right) &= 4T\left(\frac{n}{2}\right)+n^3+n^2\log n\\
&= 4\left(4T\left(\frac{n}{2^2}\right)+\left(\frac{n}{2}\right)^3+\left(\frac{n}{2}\right)^2\log\frac{n}{2}\right)+n^3+n^2\log n\\
&= 4^2T\left(\frac{n}{2^2}\right)+4\cdot\left(\frac{n}{2}\right)^3+4\left(\frac{n}{2}\right)^2\log\frac{n}{2}+n^3+n^2\log n\\
&= 4^2\left(4T\left(\frac{n}{2^3}\right)+\left(\frac{n}{2^2}\right)^3+\left(\frac{n}{2^2}\right)^2\log\frac{n}{2^2}\right)+4\cdot\left(\frac{n}{2}\right)^3+4\left(\frac{n}{2}\right)^2\log\frac{n}{2}+n^3+n^2\log n\\
&= 4^3T\left(\frac{n}{2^3}\right)+4^2\left(\frac{n}{2^2}\right)^3+4^2\left(\frac{n}{2^2}\right)^2\log\frac{n}{2^2}+4\cdot\left(\frac{n}{2}\right)^3+4\left(\frac{n}{2}\right)^2\log\frac{n}{2}+n^3+n^2\log n\\
= \ldots &= 4^kT\left(\frac{n}{2^k}\right)+\sum_{i=0}^{k-1}4^i\left(\frac{n}{2^i}\right)^3+\sum_{i=0}^{k-1}\left(4^i\left(\frac{n}{2^i}\right)^2\log\frac{n}{2^i}\right)\\
&= n^2T\left(1\right)+n^3\sum_{i=0}^{k-1}2^{-i}+n^2\sum_{i=0}^{k-1}\log\frac{n}{2^i}\\
&= n^2T\left(1\right)+n^3\sum_{i=0}^{k-1}2^{-i}+n^2\left[\log n\log n-\sum_{i=0}^{k-1}i\right]\\
&= n^2+n^3\cdot\frac{\left(\frac{1}{2}\right)^{\log n}-1}{-0.5}+n^2\log^2 n-n^2\cdot\frac{\log n}{2}\left(\log\left(n\right)-1\right)\\
&= n^2-2n^3\cdot 2^{-\log n}+2n^3+\frac{n^2}{2}\log^2\left(n\right)+n^2\cdot\frac{\log n}{2}\\
&= n^2-2n^2+2n^3+\frac{n^2}{2}\log^2\left(n\right)+n^2\cdot\frac{\log n}{2}\\
&= 2n^3+n^2\left(-1+\frac{1}{2}\log^2\left(n\right)+\frac{\log n}{2}\right)
\end{aligned}
$$

נשים לב כי $2n^3+n^2\left(-1+\frac{1}{2}\log^2\left(n\right)+\frac{\log n}{2}\right)\geq 2n^3$ לכל $n\geq 2$ ולכן עבור $c_0=2$ ו-$n_0=1$ ואז לכל $n>n_0$ מתקיים $T\left(n\right)\geq 2\cdot n^3$ ולכן $T\left(n\right)=\Omega\left(n^3\right)$.

מצד שני, מתקיים $-1+\frac{1}{2}\log^2\left(n\right)+\frac{\log n}{2}\leq n$ לכל $n\geq 4$ ולכן עבור $n_1=3$ ו-$c_1=3$ נקבל שלכל $n>n_1$ מתקיים $2n^3+n^2\left(-1+\frac{1}{2}\log^2\left(n\right)+\frac{\log n}{2}\right)\leq 2n^3+n^2\cdot n=3n^3$, כלומר ש-$T\left(n\right)=O\left(n^3\right)$.

יהי n. אזי נבחר $k=\lfloor \log n \rfloor$ כך ש-$2^k \leq n < 2^{k+1}$. נשים לב כי $T(n)$ זוהי פונקציה עולה ולכן נקבל כי

$$\begin{aligned}
T(n) &\leq T\left(2^{k+1}\right) \\
&= 2\left(2^{k+1}\right)^3 + \left(2^{k+1}\right)^2\left(-1+\frac{1}{2}\log^2\left(2^{k+1}\right)+\frac{\log 2^{k+1}}{2}\right) \\
&\leq 2^{3\log n+4} + 2^{2\log n+2}\left(-1+\frac{1}{2}\log^2\left(2^{\log(n)+1}\right)+\frac{\log 2^{\log n+1}}{2}\right) \\
&= 2^4 \cdot n^3 + 4\cdot n^2\left(-1+\frac{1}{2}\left(\log(n)+1\right)^2+\frac{\log(n)+1}{2}\right) \\
&= 2^4 \cdot n^3 + 4\cdot n^2\left(-1+\frac{1}{2}\log^2(n)+\log(n)+\frac{1}{2}+\frac{\log(n)+1}{2}\right) \\
&= 2^4 \cdot n^3 + 4\cdot n^2\left(\frac{1}{2}\log^2(n)+\frac{3\log(n)}{2}\right) \\
&\overset{(*)}{\leq} 2^4 \cdot n^3 + 4\cdot n^2 \cdot \left(\frac{1}{2}n+\frac{3n}{2}\right) \\
&= 2^4 \cdot n^3 + 8\cdot n^3 \\
&= 24n^3
\end{aligned}$$

הסבר ל-$(*)$ - נשים לב לכל $n \geq 2$ מתקיים $\frac{1}{2}\log^2(n) \leq \frac{1}{2}n$ ובנוסף $\frac{3\log(n)}{2} \leq \frac{3n}{2}$.

כלומר, קיבלנו כי עבור $c=24$ ו-$n_0=1$ מתקיים שלכל $n>n_0$: $T(n) \leq 24n^3$ ולכן $T(n)=O\left(n^3\right)$ לכל n כללי.

מצד שני, מתקיים $T(n)=4T\left(\frac{n}{2}\right)+n^3+n^2\log n \geq n^3$ ולכן עבור $n_1=1$ ו-$c_1=1$ נקבל שלכל $n>n_1$ מתקיים $T(n) \geq n^3$ ולכן $T(n)=\Omega\left(n^3\right)$.

סה"כ קיבלנו כי $T(n)=\Theta\left(n^3\right)$.

סעיף ב2

מתקיים:

$$\begin{aligned}
\log\left((n^2)!\right) &= \log 1 + \log 2 + \cdots + \log\left(n^2\right) \\
&\geq \log\left(\frac{n^2}{2}\right) + \log\left(\frac{n^2}{2}\right) \cdots + \log\left(\frac{n^2}{2}\right) \\
&\geq \frac{n^2}{2}\log\left(\frac{n^2}{2}\right) \\
&= \frac{n^2}{2}\cdot\left(\log n^2 - \log 2\right) \\
&= n^2\log n - \frac{n^2}{2} \\
&= \frac{n^2}{2}\log n + \frac{n^2}{2}\log n - \frac{n^2}{2} \\
&= \frac{n^2}{2}\log n + \frac{n^2}{2}\left(\log n - 1\right) \\
&\geq \frac{n^2}{2}\log n
\end{aligned}$$

כאשר בצעד האחרון השתמשנו בעובדה שלכל $n \geq 2$ הביטוי $(\log n - 1) \geq 0$.

לכן עבור $n_0 = 1$ ו $c = \frac{1}{2}$ נקבל ש-$\log\left((n^2)!\right) \geq \frac{n^2}{2}\log n$ ולכן $\log\left((n^2)!\right) = \Omega\left(n^2\log n\right)$.

מצד שני, מתקיים

$$\begin{aligned}
\log\left((n^2)!\right) &= \log 1 + \log 2 + \cdots + \log\left(n^2\right) \\
&\leq \log\left(n^2\right) + \log\left(n^2\right) \cdots + \log\left(n^2\right) \\
&= n^2\log\left(n^2\right) \\
&= 2n^2\log\left(n\right)
\end{aligned}$$

לכן, עבור $n_1 = 1$ ו-$c_1 = 2$ נקבל שלכל $n > n_1$ מתקיים $\log\left((n^2)!\right) \leq 2n^2\log\left(n\right)$ ולכן $\log\left((n^2)!\right) = O\left(n^2\log n\right)$.

סה"כ, קיבלנו כי $\log\left((n^2)!\right) = \Theta\left(n^2\log n\right)$.

סעיף ב3

נציב ערכים מהצורה: $n = 2^{2^k}$

מתקיים:

$$
\begin{aligned}
T\left(2^{2^k}\right) &= T\left(2^{2^{k-1}}\right) + 2^{2^{k+1}} \\
&= T\left(2^{2^{k-2}}\right) + 2^{2^k} + 2^{2^{k+1}} \\
&= T\left(2^{2^{k-3}}\right) + 2^{2^{k-1}} + 2^{2^k} + 2^{2^{k+1}} \\
= \ldots &= T\left(2\right) + \sum_{i=0}^{k-1} 2^{2^{k-i+1}} \\
&= T\left(\lfloor\sqrt{2}\rfloor\right) + 4 + \sum_{i=0}^{k-1} 2^{2^{k-i+1}} \\
&= 1 + 4 + 2^{2^{k+1}} \cdot \sum_{i=0}^{k-1} 2^{2^{-i}}
\end{aligned}
$$

נציב ערכים מהצורה: $n = 2^k$

מתקיים:

$$
\begin{aligned}
T\left(2^k\right) &= T\left(2^{\frac{k}{2}}\right) + 2^{2k} \\
&= T\left(2^{\frac{k}{2^2}}\right) + 2^k + 2^{2k} \\
&= T\left(2^{\frac{k}{2^3}}\right) + 2^{\frac{k}{2}} + 2^k + 2^{2k} \\
= \ldots &= T\left(2\right) + \sum_{i=0}^{\log k} 2^{\frac{2k}{2^i}} \\
&= T\left(2\right) + \sum_{i=0}^{\log k} 2^{\frac{2k}{2^i}} \\
&= T\left(1\right) + 2^{k+1} \cdot \frac{\left(\frac{1}{2}\right)^{k-1} - 1}{\frac{1}{2} - 1} \\
&= T\left(1\right) - 2^{k+2} \cdot \frac{1}{2^{k-1}} + 2^{k+2} \\
&= T\left(1\right) - 2 + 2^{k+2}
\end{aligned}
$$

נגדיר $S\left(k\right) = T\left(2^k\right)$. מתקיים:

$$
S\left(k\right) = S\left(\frac{k}{2}\right) + 2^{2k}
$$

כעת, נשתמש בשיטת המאסטר השלישית.

נסמן: $a=1,\ b=2,\ \log_{b}\left(a\right)=\log_{2}\left(1\right)=0,\ f\left(n\right)=f\left(2^{k}\right)=2^{2k}=n^{2}$.

נבחר $\varepsilon=1$. נשים לב כי $f\left(n\right)=\Omega\left(n\right)$ מכיוון שעבור $c=1$ ואז עבור $n_{0}=0$ לכל $n>n_{0}$ מתקיים $n^{2}\geq c\cdot n$.

בנוסף, מתקיים עבור $c=\frac{1}{2}$:

$$1\cdot f\left(\frac{n}{2}\right)=\left(\frac{n}{2}\right)^{2}\leq\frac{1}{2}\cdot n^{2}=\frac{1}{2}\cdot f\left(n\right)$$

.

לכן, נסיק משיטת המאסטר כי $T\left(n\right)=\Theta\left(n^{2}\right)$.

סעיף ב4

נסמן $S\left(n\right)=\frac{T(n)}{n^{8}}$. ואז מתקיים:

$$\begin{aligned}T\left(n\right)&=16n^{4}T\left(\sqrt{n}\right)+2n^{8}\log^{4}n\\ \Longrightarrow S\left(n\right)&=\frac{16n^{4}T\left(\sqrt{n}\right)+2n^{8}\log^{4}n}{n^{8}}=\frac{16T\left(\sqrt{n}\right)}{n^{4}}+2\log^{4}n\\ T\left(\sqrt{n}\right)&=S\left(\sqrt{n}\right)\cdot\left(n^{0.5}\right)^{8}=S\left(\sqrt{n}\right)\cdot n^{4}\end{aligned}$$

$$S\left(n\right)=16\cdot S\left(\sqrt{n}\right)+2\log^{4}n$$

נסמן $n=2^{m}$ ובנוסף $P\left(m\right)=S\left(n\right)=S\left(2^{m}\right)$. מתקיים:

$$\begin{aligned}P\left(m\right)&=16\cdot S\left(2^{\frac{m}{2}}\right)+2m^{4}\\ &=16\cdot P\left(\frac{m}{2}\right)+2m^{4}\end{aligned}$$

נשתמש בשיטת המאסטר השנייה.

מתקיים $a=16$. $b=2$. לכן $\log_{b}a=4$, $f\left(m\right)=2m^{4}$.

מתקיים $m^{4}\leq2m^{4}\leq3\cdot m^{4}$ ולכן עבור $c_{1}=3$ מתקיים ש-$f\left(m\right)=O\left(m^{4}\right)$ ועבור $c_{2}=1$ מתקיים ש-$f\left(m\right)=\Omega\left(m^{4}\right)$ לכל m. לכן $f\left(m\right)=\Theta\left(m^{\log_{b}a}\right)=\Theta\left(m^{4}\right)$.

לכן, נסיק משיטת המאסטר השנייה כי מתקיים: $P\left(m\right)=\Theta\left(m^{4}\cdot\log m\right)$.

מהצבה חוזרת נקבל כי $S\left(n\right)=P\left(m\right)=\Theta\left(m^{4}\cdot\log m\right)=\Theta\left(\log^{4}n\cdot\log\log n\right)$.

מהצבה אחרונה וחביבה נקבל כי $T\left(n\right)=S\left(n\right)\cdot n^{8}=\Theta\left(n^{8}\cdot\log^{4}n\cdot\log\log n\right)$

נוכיח כי $T\left(n\right)=\Theta\left(n^{8}\cdot\log^{4}n\cdot\log\log n\right)$.

נתון כי $S\left(n\right)=\Theta\left(\log^{4}n\cdot\log\log n\right)$. לכן קיימים $c_{1},c_{2}>0$ ו-$n_{1},n_{2}>0$ כך שלכל $n>n_{1},n_{2}$ מתקיים $c_{1}\cdot\log^{4}n\cdot\log\log n\leq S\left(n\right)\leq c_{2}\cdot\log^{4}n\cdot\log\log n$.

לאחר הכפלה ב-n^{8} של אי השיוויון נקבל כי לכל $n>\max\left\{n_{1},n_{2}\right\}$ מתקיים

$$c_{1}\cdot n^{8}\cdot\log^{4}n\cdot\log\log n\leq S\left(n\right)\cdot n^{8}\leq c_{2}\cdot n^{8}\cdot\log^{4}n\cdot\log\log n$$

כלומר, קיבלנו כי עבור $c_{1}'=c_{1}$ ו-$n_{1}'=n_{1}$ מתקיים ש-$T\left(n\right)=\Omega\left(n^{8}\cdot\log^{4}n\cdot\log\log n\right)$.

בנוסף, קיבלנו כי עבור $c_{2}'=c_{2}$ ו-$n_{2}'=n_{2}$ מתקיים ש-$T\left(n\right)=O\left(n^{8}\cdot\log^{4}n\cdot\log\log n\right)$.

סה"כ, קיבלנו כי $T\left(n\right)=\Theta\left(n^{8}\cdot\log^{4}n\cdot\log\log n\right)$.

שאלה 4

ראשית, נניח לצורך נוחות כי האינדקס במערך $State$ מתחילים מ-1.

מבנה הנתונים שלנו יכיל את המידע הבא –

$states$ - מערך בגודל N (לפי מספר המדינות) המתחיל מאינדקס 1 עד N אשר שומר אובייקט מסוג $state\_node$ המכיל את המשתנים הבאים :

1. $int\ id$ - מספר המזהה של המדינה
2. $int\ \ const\_quarantine$ - מספר המסמל את קבוע הסגר.
3. $close\_node * ptr$ - מצביע לאובייקט מטיפוס $close\_node$.

בנוסף, $close\_node$ יכיל את המשתנים הבאים :

1. $state\_node * ptr$ - מצביע לאובייקט מטיפוס $state\_node$.
2. $int\ column$ - משתנה אשר שומר את המיקום במערך $Close$ שבו האובייקט נמצא.

$Close$ - מערך בגודל D שמכיל רשימות המקיימות את התכונה הבאה : ב-$close\,[i]$ קיימת רשימה של מדינות שכולן נכנסות לסגר עוד $(i-day)_{mod(D)}$ ימים.

למשל, אם $day\ =\ 1$ ו-$i\ =\ 3$ אז המדינות שנמצאות ב-$close\,[3]$ ייכנסו לסגר בעוד יומיים. כאשר נקרא ל-$lock\_down\_now\,()$ פעמיים, אזי day יגדל ב-2 (הסבר נוסף למטה) ו-$day\ \ =\ \ 3$ ובפונקציה זו אנו מדפיסים את הרשימה הנמצאת ב-$close\,\left[day_{mod(D)}\right]$, ואכן אלו המדינות שצריכות להיות מודפסות בעוד יומיים.

הסבר על המשתנה $column$: $column$ מסמל את המיקום במערך $close$ שבו המדינה נמצאת. הסיבה שאנו שומרים אותו היא מכיוון שזה מאפשר לנו להעביר אובייקט מטיפוס $close\_node$ מרשימה בתא $close\,[j]$ לרשימה אחרת ב-$close\,[k]$ ב-$O\,(1)$.

כלומר, אם מדינה נמצאת ברשימה ב-$close\,[k]$ אז היינו רוצים לדעת מתוך איבר ברשימה היכן הרשימה ששייכת אליו נמצאת במערך.

הסבר :

$\underline{Init(N,D)}$ - ניצור את מבנה הנתונים שתיארנו לעיל. כלומר, ניצור מערך $states$ מסוג $state\_node$ ומערך $close$ מסוג $List\,\langle close\_node\rangle$ ובנוסף נאתחל משתנה day להיות אפס המהווה את היום הנוכחי.

$\underline{Insert\,(i, D_i, days)}$ - במידה וקיבלנו מספר מדינה i שכבר נמצא במערך $states$ נחזיר הודעת שגיאה ולא יתבצע כלום (בדיקה זו ב-$O\,(1)$ - בהינתן i ניגש לתא ה-i במערך $states$ ונבדוק אם הוא מאותחל או לא).

אחרת, ניצור אובייקט מסוג $state\_node$ בשם new_state_node ונתאחל את הערכים הבאים :

1. $new\_state\_node \to id = i$
2. $new\_state\_node \to const\_quarantine = D_i$
3. נשמור את האיבר במקום ה-i במערך $states$.

לאחר מכן, ניצור אובייקט מסוג $close\_node$ בשם new_close_node ונבצע את הפעולות הבאות:

1. $new\_close\_node \to ptr = \&new\_state\_node$
2. $new\_close\_node \to column = (days + day)_{mod(D)}$

ניגש ל-$close\left[(days+day)_{mod(D)}\right]$ ונבדוק האם התא מאותחל. במידה והתא מאותחל נוסיף את new_close_node לתחילת הרשימה, אחרת ניצור רשימה ריקה ונוסיף את new_close_node לתחילת הרשימה.

כמו כן, נגדיר את $new\_state\_node \to ptr = \&new\_close\_node$.

$\underline{Precede\_lockdown(i,\ days)}$ - ראשית, נשים לב כי לא ניתן להקדים את הסגר של המדינה ביותר ממספר הימים שנותר לה לסגר הקרוב. לכן, נבדוק האם $days < ((states\,[i] \to ptr \to column) - day)_{mod(D)}$ - אם זה אכן קורה, נחזיר הודעת שגיאה.

אחרת, ניגש לאיבר $states\,[i] \to ptr$ ונמחק אותו מהרשימה בה הוא נמצא (זוהי רשימה דו-כיוונית ולכן ניתן להסיר ב-$O\,(1)$) ונכניס אותו לתחילת הרשימה ב-$close\left[((states\,[i] \to ptr \to column) - days)_{mod(D)}\right]$.

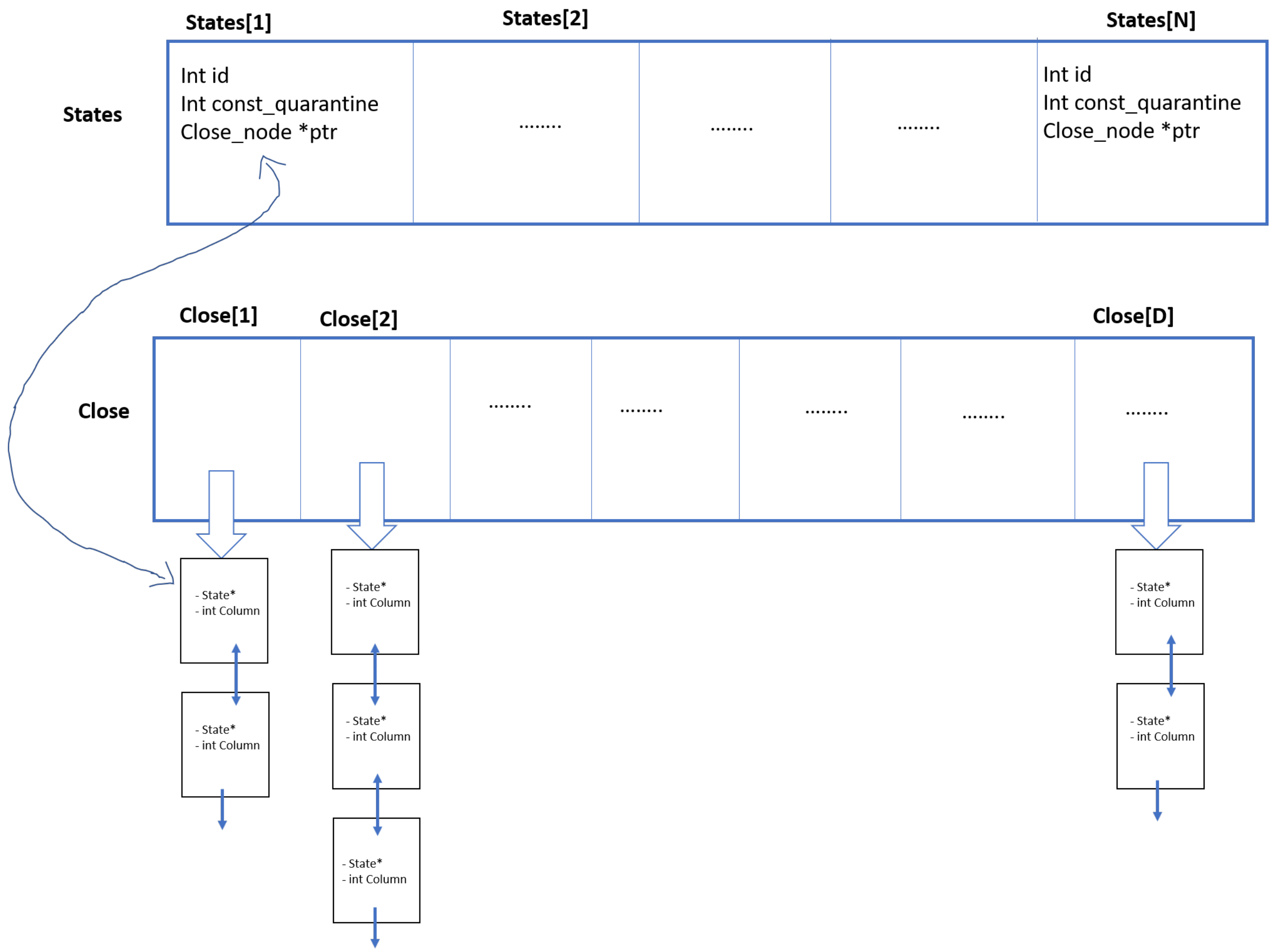
בנוסף, נשנה את $(states\,[i] \to ptr \to column)$ לערך $((states\,[i] \to ptr \to column) - days)_{mod(D)}$.

לדוגמא, אם $D = 5$ וישנה מדינה עם $id = 1$ אשר נמצאת ב-$close\,[4]$ ו-$day = 1$ אז אנו יודעים כי בעוד 3 ימים מדינה זו נכנסת לסגר, אך בעקבות מצב חירום נקדים ביום את הסגר נצטרך להעביר מדינה זו מהרשימה ב-$close\,[4]$ לרשימה ב-$close\,[3]$ כדי שבעוד יומיים כשהיא תיכנס לסגר היא תודפס.

$\underline{lock\_down\_now()}$ - ראשית, נבצע $day++$ (על-מנת לסמן שעבר יום) ולאחר מכן ניגש ל-$close\left[day_{mod(D)}\right]$ ונבצע הדפסה של הרשימה השמורה בתא זה, כלומר לכל $element$ ברשימה נדפיס את $element \to ptr \to id$.

נעבור שוב על הרשימה ועבור כל $element$ ברשימה נוציא אותו ונכניס אותו לתחילת הרשימה ב-$close\left[(day + (element \to ptr \to const\_quarantine))_{mod(D)}\right]$.

לדוגמא: כאשר הגענו ליום x בו המדינה y נכנסת לסגר - לכן הפעם הבאה שהיא תהיה בסגר תהיה עוד $const\_quarantine$ ימים אבל מכיוון שאנו שומרים את המערך באופן ציקלי נבצע $modD$.


States[1]
States[2]
States[N]
States
Int id
Int const_quarantine
Close_node *ptr
....
....
....
Int id
Int const_quarantine
Close_node *ptr
Close[1]
Close[2]
Close[D]
Close
....
....
....
....
....
- State*
- int Column
- State*
- int Column
- State*
- int Column
- State*
- int Column
- State*
- int Column
- State*
- int Column
- State*
- int Column

שאלה 5

ראשית, נניח לצורך נוחות כי האינדקס במערך $State$ מתחילים מ-1.

מבנה הנתונים שלנו יכיל את המידע הבא -

$States$ - מערך המכיל N איברים מאינדקס 1 ועד N כאשר כל איבר מטיפוס $state\_node$.

האובייקט $state\_node$ יכיל את המשתנים הבאים :

1. $int\ id$ - מזהה מדינה.
2. $int\ num\_of\_arrivals$ - מספר האנשים שרוצים להגיע למדינה.
3. $int\ num\_of\_departures$ - מספר האנשים שרוצים לטוס מהמדינה.
4. $vote\_node * \ ptr$ - מצביע לאובייקט מטיפוס $vote\_node$.

$votes$ - רשימה דו-כיוונית מטיפוס $votes\_struct$ כאשר $votes\_struct$ יכיל את המשתנים הבאים :

1. $int\ score$- משתנה המכיל את מספר ההצבעות שהצביעו למדינות הנמצאות ברשימה.
2. $elements\_in\_vote\ \ List\langle vote\_node\rangle$ - רשימה דו-כיוונית מטיפוס $vote\_node$ המכיל מדינות ש-$score$ אנשים רוצים לטוס אליהן.

$vote\_node$ - אובייקט המכיל את המשתנים הבאים :

1. $int\ score$ - משתנה המסמל את מספר האנשים שרוצים לטוס למדינה זו.
2. $state\_node * \ ptr$ - מצביע ל-$state\_node$.
3. $votes\_struct * \ head$ - מצביע לאיבר ברשימה $votes$ מטיפוס $votes\_struct$ שמחזיק רשימה של $vote\_node$ (כמובן שה-$score$ של כל המדינות ברשימה זו שווה ל-$head \to score$).

$last\_node$ - מצביע מטיפוס $votes\_struct$ לאיבר האחרון ב-$votes$.

$unwanted$ - מערך המכיל N איברים מאינדקס 1 ועד N כאשר כל איבר מטיפוס $unwanted\_node$ המכיל את המשתנים הבאים :

1. $int\ check$ - מזהה מדינה.
2. $int\ forward\_id$ - האינדקס של המדינה העוקבת במערך שאף אדם עדיין לא רוצה לטוס אליה.
3. $int\ back\_id$ - האינדקס של המדינה הקודמת במערך שאף אדם עדיין לא רוצה לטוס אליה.

$int\ head\_unwanted$ - המדינה הראשונה במערך $unwanted$ שאף אדם לא רוצה לטוס אליה.

הסבר:

$Init\left(N\right)$:

נאתחל את $State$ בגודל N המכיל את N המדינות, את הרשימה $votes$ ובנוסף נאתחל את $votes\_struct$ להצביע ל-$null$.

בנוסף, נאתחל את המערך $unwanted$ כאשר לכל $2 \leq i \leq N-1$ יתקיים $unwanted[i] \rightarrow forward\_id = i+1$ ו-$unwanted[i] \rightarrow back\_id = i-1$

ובנוסף $unwanted[1] \rightarrow back\_id = -1$ ובנוסף $unwanted[n] \rightarrow forward\_id = -1$.

לבסוף, נאתחל את $head\_unwanted$ ל-1.

בנוסף, נאתחל $unwanted[i] \rightarrow check = 1$ לכל $1 \leq i \leq N$.

מההרצאה אנו יודעים כי ניתן לאתחל מערך ב-$O\left(1\right)$.

$Fly\,(j,i)$:

אם $unwanted[i] \to check = -1$ אז לא נשנה את האיברים ב-$unwanted$.

אחרת,

- אם $i = head\_unwanted$ וגם $unwanted[i] \to forward\_id = -1$ נבצע את הפעולה הבאה:

1. $unwanted[i] \to check = -1$.
2. $head\_unwanted = -1$.

- אחרת, אם $i = head\_unwanted$ נבצע את הפעולה הבאה:

1. $unwanted\,[(unwanted[i] \to forward\_id)] \to back\_id = unwanted[i] \to back\_id$
2. $head\_unwanted = unwanted[i] \to forward\_id$
3. $unwanted[i] \to check = -1$

- אחרת, נבדוק האם $unwanted[i] \to forward\_id$ שווה ל-1-.

אם כן, נבצע את הפעולה הבאה:

1. $unwanted\,[unwanted\,[i] \to back\_id] \to forward\_id = unwanted[i] \to forward\_id$
2. $unwanted[i] \to check = -1$

אחרת, נבצע את 3 הפעולות הבאות:

1. $unwanted\,[(unwanted[i] \to forward\_id)] \to back\_id = unwanted[i] \to back\_id$.
2. $unwanted\,[unwanted\,[i] \to back\_id] \to forward\_id = unwanted[i] \to forward\_id$
3. $unwanted[i] \to check = -1$

אם $State[i]$ לא מאותחל - ניצור אובייקט מטיפוס $state\_node$ ונשמור אותו ב-$State[i]$ ונאתחל את שדותיו באופן הבא:

1. $State[i] \to id = i$
2. $State[i] \to num\_of\_arrivals = 0$
3. $State[i] \to num\_of\_departures = 0$
4. $State[i] \to ptr = null$

אם $State[j]$ לא מאותחל - ניצור אובייקט מטיפוס $state\_node$ ונשמור אותו ב-$State[j]$ ונאתחל את שדותיו באופן הבא:

1. $State[j] \to id = j$

2. $State[j] \to num\_of\_arrivals = 0$

3. $State[j] \to num\_of\_departures = 0$

4. $State[i] \to ptr = null$

כעת נבצע את הפעולות :

1. $(State[i] \to num\_of\_arrivals) + +$

2. $(State[j] \to num\_of\_departures) + +$

- במידה ו-$State[i] \to ptr = null$ ניצור אובייקט מטיפוס $vote\_node$ בשם new_vote_node ונבצע את הפעולות הבאות :

1. $new\_vote\_node \to score = 1$

2. $new\_vote\_node \to ptr = State[i]$

אם $votes$ הוא לא רשימה ריקה וגם $votes \to score == 1$ - אם כן, נשרשר את new_vote_node לתחילת הרשימה ב-$votes \to$ $.elements\_in\_vote$ במידה ו-$votes \to next == null$ אזי ברשימה $votes$ יש איבר יחיד ואז המצביע $last\_node$ ישמור את הכתובת של $.votes$

אחרת, ניצור אובייקט מטיפוס $votes\_struct$ בשם new_votes_struct כאשר נבצע את הפעולות הבאות :

1. $.new\_votes\_struct \to score = 1$

2. ניצור את הרשימה $.new\_votes\_struct \to \ elements\_in\_vote$

3. נוסיף את new_vote_node לתחילת הרשימה $.new\_votes\_struct \to \ elements\_in\_vote$

4. במידה וה-$votes$ היא רשימה ריקה המצביע $last\_node$ ישמור את הכתובת של $.new\_votes\_struct$

5. נשרשר את new_votes_struct לתחילת הרשימה $.votes$

במידה ו-• לא מתקיים נבצע את הפעולות הבאות :

1. $.state[i] \to ptr \to score + +$

2. נוציא את $state[i] \ \to \ ptr$ מהרשימה בה הוא נמצא כרגע. אם זהו האיבר האחרון שהיה ברשימה נסיר את $state[i] \ \to$ $ptr \ \to \ head$ מהרשימה $votes$ (ניתן להסיר אותו ב-$O\left(1\right)$ כי זוהי רשימה דו-כיוונית). נסתכל על $state[i] \ \to \ ptr \ \to$ $.head \to next \to score$ אם מספר זה שווה ל-$state[i] \to ptr \to score$ אז נוסיף את $(State[i] \to ptr) *$ לתחילת הרשימה של האיבר הבא, דהיינו ב-$.state[i] \to ptr \to head \to next \to elements\_in\_vote$ בנוסף, נשנה את הערך של $state[i] \to ptr \to head$ להיות $.state[i] \to ptr \to head \to next$

אחרת, ניצור אובייקט מטיפוס $votes\_struct$ בשם new_votes_struct ונבצע את הפעולות הבאות:

1.2 $new\_votes\_struct \to score = state[i] \to ptr \to score$.

2.2 ניצור רשימה חדשה ב-$new\_votes\_struct \to elements\_in\_vote$ ונשרשר לתחילתה את $(State[i] \to ptr)*$.

3.2 נשרשר את האיבר new_votes_struct בין האיבר $state[i] \to ptr \to head$ לבין $state[i] \to ptr \to head \to next$.

4.2 נשנה את הערך של $state[i] \to ptr \to head$ להיות $state[i] \to ptr \to head \to next$.

אם $state[i] \to ptr \to head \to next == null$ - דהיינו הגענו לאיבר האחרון ברשימה $votes$ ואז $last\_node$ ישמור את הכתובת של $state[i] \to ptr \to head \to next$.

$Arrivals\,(i)$:

נבדוק אם קיים אובייקט מטיפוס $state\_node$ ב-$State[i]$.

אם כן - נחזיר את $State[i] \to num\_of\_arrivals$.

אחרת, נחזיר 0.

$Departures(j)$:

נבדוק אם קיים אובייקט מטיפוס $state\_node$ ב-$State[i]$.

אם כן - נחזיר את $State[i] \to num\_of\_departures$.

אחרת, נחזיר 0.

$Favored\,(k)$:

ניגש לערך של $last\_node$ - זוהי אובייקט מסוג $votes\_struct$ - זהו האובייקט האחרון ברשימה $vote$ (בעל ה-$score$ המקסימלי).

אם יש לפחות k איברים, נדפיס את k האיברים הראשונים. אחרת, נדפיס את האיברים ברשימה $last\_node \to elements\_in\_vote$ ונעבור לרשימה בעלת הדירוג הנמוך (זהו האיבר הקודם של של $last\_node$ ברשימה $votes$, ככה נמשיך באינדוקציה עד שנגיע להדפסת k איברים.

$Avoided\,()$:

ראשית, נבדוק האם $head\_unwanted = -1$. במידה וכן, לא נדפיס כלום.

אחרת, ניגש ל-$unwanted\,[head\_unwanted]$ ונדפיס את $unwanted\,[head\_unwanted] \to id$.

נעבור לתא הבא במערך באמצעות שדה ה-$forward\_id$ - נדפיס את $unwanted\,[head\_unwanted \to forward\_id] \to id$ וככה נמשיך להדפיס את כל המדינות עד שנגיע למדינה האחרונה שבה השדה $forward\_id$ הוא -1.