מבני נתונים

מטלה 1

שאלה 1: הוכח או הפרך:

טענה 1:

$$\begin{cases} f(n) = (\sqrt{n})^{\sqrt{n}} \\ g(n) = \lfloor \sqrt{n} \rfloor ! \end{cases} \Longrightarrow \log(f(n)) = \Theta(\log(g(n)))$$

הטענה נכונה: הוכחה:

נציב: $x = \sqrt{n}$:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{(\sqrt{n})^{\sqrt{n}}}{\lfloor \sqrt{n} \rfloor !} = \lim_{x \to \infty} \frac{x^x}{x!} = \infty$$

לכן $f(n) = O(g(n))$ ו- $g(n) = \Omega(f(n))$

אבל $f(n) \neq \Omega(g(n))$ ולכן $f n \neq \Theta(g(n))$

ולכן $\log(f(n)) \neq \Theta(\log(g(n)))$.

טענה 2:

$$T(n) = \left(T\left(\tfrac{n}{2}\right)\right)^2 \cdot 2^n \quad , \quad T(1) = 2$$

אז: $T(n) = 2^{\Theta(n)}$

$$\begin{aligned}
T(n) &= \left(T\left(\tfrac{n}{2}\right)\right)^2 \cdot 2^n \\
&= \left(T\left(\tfrac{n}{4}\right)^2 \cdot 2^{\frac{n}{2}}\right) \cdot 2^n \\
&= \left(T\left(\tfrac{n}{4}\right)^2\right) \cdot 2^{\frac{3n}{2}} \\
&= \left(T\left(\tfrac{n}{8}\right)^2 \cdot 2^{\frac{n}{4}}\right) \cdot 2^{\frac{3n}{2}} \\
&= \left(T\left(\tfrac{n}{8}\right)^2\right) \cdot 2^{\frac{7n}{2}} = \left(T\left(\tfrac{n}{2^k}\right)^2\right) \cdot 2^{\frac{kn}{2^k}}
\end{aligned}$$

עבור $K = \log_2(n)$: $= \left(T(1)^2\right) \cdot 2^{n \log_2 n}$

אבל $T(1) = 2$, לכן: $= (2^2) \cdot 2^{n \log_2 n} = \boxed{4n^n}$

$\boxed{\text{ולכן עבור } n = c: \ T(n) = 2^{\Theta(n)}}$

שאלה 3:

$$g(n) = O(n) \Leftarrow f(n) = \Omega(n) \ , \ f(g(n)) = O(n)$$

(1) $$f(m) = \Omega(m) \Leftrightarrow f(m) \geq cm \ , \ \forall m \geq m_0$$

עבור c קבוע גדול מאפס ו- $m_0 \in \mathbb{N}^+$.

(2) $$f(g(n)) = O(n) \Leftrightarrow f(g(n)) \leq kn \ , \ \forall n \geq n_0$$ מכוס g,

עבור k קבוע גדול מאפס ו- $n_0 \in \mathbb{N}^+$.

לכן, (1) $$f(m) \geq cm \Rightarrow f(g(n)) \geq c \cdot g(n)$$

לכן

$$c \cdot g(n) \leq f(g(n)) \leq k \cdot n$$

ולכן

$$g(n) = O(f(g(n))) = O(n)$$

■

שאלה 2 : סידור מהקטנה לגדולה :

$f_2 > f_6 > f_5 > f_4 > f_7 > f_1 > f_3$

כלומר:

$$2^{3^n} > n^n > 4^n > 2^{\log_{\sqrt{2}}(n)} > \log_3(3^n \cdot n^2) > 2019 > \frac{1}{n^{2020}}$$

הוכחה:

• $Of_2(n) = f_6(n)$:

$$\lim_{n\to\infty} \frac{f_2(n)}{f_6(n)} = \lim_{n\to\infty} \frac{2^{3^n}}{n^n} = \lim_{n\to\infty} \frac{\log(2^{3^n})}{\log(n^n)} = \lim_{n\to\infty} \frac{3^n \cdot \log(2)}{n\log(n)}$$

$$= \log(2) \cdot \lim_{n\to\infty} \frac{3^n}{n\log(n)} \overset{\frac{'\infty'}{'\infty'}}{=} \log(2) \cdot \lim_{n\to\infty} \frac{(3^n)'}{(n\log(n))'} \quad (*)$$

• $(a^x)' = a^x \cdot \ln(a) \Rightarrow (3^n)' = 3^n \log(3)$

• $(n\log(n))'$: $f(n) = n,\ g(n) = \log(n),\ (f \cdot g)' = f'g + fg' = \log(n) + \frac{1}{n} \cdot n = \log(n) + 1$

$$(*) = \log(2) \cdot \lim_{n\to\infty} \frac{3^n \log(3)}{\log(n)+1} \overset{\frac{'\infty'}{'\infty'}}{=} \log(2) \cdot \lim_{n\to\infty} \frac{3^n \cdot \log^2(3)}{\frac{1}{n}} = \log(2) \cdot \log^2(3) \cdot \lim_{n\to\infty} \frac{3^n}{\frac{1}{n}} = \infty$$

לכן $Of_2(n) = f_6(n)$ •

• $Of_6(n) = f_5(n)$:

$$\lim_{n\to\infty} \frac{f_6(n)}{f_5(n)} = \lim_{n\to\infty} \frac{n^n}{4^n} = \lim_{n\to\infty} \frac{\log(n^n)}{\log(4^n)} = \lim_{n\to\infty} \frac{\cancel{n}\log(n)}{\cancel{n}\log(4)}$$

$$= \frac{1}{\log(4)} \cdot \lim_{n\to\infty} \log(n) = \infty$$

לכן $Of_6(n) = f_5(n)$ •

• $Of_5(n) = f_4(n)$:

$f_4(n) = 2^{\log_{\sqrt{2}}(n)}$. לפי הנוסחא $c^{\log_c(n)} = n$

$$f_4(n) = (\sqrt{2})^{2\log_{\sqrt{2}}(n)} = (\sqrt{2})^{\log_{\sqrt{2}}(n^2)} = n^2$$

מספיק :

$$\lim_{n\to\infty} \frac{f_5(n)}{f_4(n)} = \lim_{n\to\infty} \frac{4^n}{n^2} \overset{\frac{\infty}{\infty}}{=} \lim_{n\to\infty} \frac{4^n \ln(4)}{2n} = \ln(4) \cdot \lim_{n\to\infty} \frac{4^n \ln(4)}{2} = \infty$$ לכן:

$\blacksquare\ O f_5(n) = f_4(n)$ לכן

$$\lim_{n\to\infty} \frac{f_4(n)}{f_7(n)} = \lim_{n\to\infty} \frac{n^2}{\log_3(3^n \cdot n^2)} = \lim_{n\to\infty} \frac{n^2}{\log_3(3^n) + \log_3(n^2)}$$

$:O\ f_4(n) = f_7(n)\ \bullet$

$$= \lim_{n\to\infty} \frac{n^2}{n\log_3(3) + 2\log_3(n)} = \lim_{n\to\infty} \frac{n^2}{n + 2\log_3(n)} = \lim_{n\to\infty} \frac{n}{\frac{1}{n} + \frac{2\log_3(n)}{n}} = \infty$$

$\blacksquare\ O f_4(n) = f_7(n)$ לכן

$:O f_7(n) = f_1(n)\ \bullet$

$$\lim_{n\to\infty} \frac{\log_3(3^n \cdot n^2)}{2019^{2020}} = \lim_{n\to\infty} \frac{\log_3(3^n) + \log_3(n^2)}{2019^{2020}}$$

$$= \lim_{n\to\infty} \frac{n\log_3(3) + 2\log_3(n)}{2019^{2020}} = \frac{1}{2019^{2020}} \cdot \lim_{n\to\infty} n + 2\log_3(n) = \infty$$

$\blacksquare\ O f_7(n) = f_1(n)$ לכן

$$\lim_{n\to\infty} \frac{2019^{2020}}{\frac{1}{n^{2020}}} = 2019^{2020} \cdot \lim_{n\to\infty} n^{2020} = \infty$$

$:O f_1(n) = f_3(n)\ \bullet$

$\blacksquare\ O f_1(n) = f_3(n)$ לכן

שאלה 3:

A: כדי שהלולאה תרוץ פעם אחת בלבד נצטרך ש $n = 2^3$

כדי שהיא תרוץ פעמיים נצטרך $n = 2^{3^1}$

כדי שהיא תרוץ k פעמים נצטרך ש- $n = 2^{3^k}$

$$n = 2^{3^k} \iff \log(n) = \log(2^{3^k}) \iff \log(n) = k\log(2^3)$$

$$\iff k = \frac{\log(n)}{\log(2^3)}$$

$$\iff k = \log(n-2^3) \implies O(\log(n))$$

B: עבור $i=1$ הלולאה for(j=1, j<=i, j++) תרוץ פעם אחת.

עבור $i=2$ היא תרוץ פעמיים.

עבור $i=n$ היא תרוץ n פעמים.

לכן, סך הכל הלולאה הנמצאת תרוץ $1+2+3+\dots+n = \frac{n(n+1)}{2}$ פעמים.

ועל פעם היא קוראת לפונקציה mystery1(n) שסיבוכיותה שווה ל- $O(\log(n))$.

לכן נקבל: $\log(n) + 2\log(n) + 3\log(n) + \dots + n\log(n)$

$$\sum_{i=1}^{n} i\log(n) = \log(n)\cdot\sum_{i=1}^{n} i = \log(n)\cdot\frac{n(n+1)}{2} = O(n^2\log(n)).$$

C: כל עוד ש- $x<i$ מכפילים את x פי 2. לכן צריך $\log(i)$ צעדים כדי לעבור את i. לכן, החלק הראשון של הקוד שווה ל- $O(\log(i))$.

בחלק השני x עבר את i, ולכן $x \geq i$

כל עוד ש- $i>2$ נפעיל עליו שורש 2.

כדי שזה ירוץ פעם אחת בלבד, צריך $i = 2^2$

כדי שזה ירוץ פעמיים צריך $i = 2^{2^2}$

כדי שזה ירוץ K פעמים צריך $i = 2^{2^k}$

$\Leftarrow k = \log(i-2^2) = O(\log(i))$ ולכן קיבלנו סה"כ $O(\log(i)+\log(i)) = O(\log(i))$

ד: הקוד מריץ $mystery2(n)$, ואז $mystery2(n-1)$, וכו' עד $mystery2(1)$.

כמו שראינו ב-ג, $O(mystery2(i)) = \log(i)$, ולכן:

$$O(mystery2(n) + mystery2(n-1) + \dots + mystery2(1)) = O(\log(n) + \log(n-1) + \dots + \log(1))$$

$$\log(n) + \log(n-1) + \dots + \log(1) = \log(n \times (n-1) \times \dots \times 1) = \log(n!)$$

ולפי נוסחת סטירלינג: $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$

$$\Leftarrow \log(n!) = \log\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)$$
$$= \log(\sqrt{2\pi n}) + \log\left(\frac{n}{e}\right)^n$$
$$= n\log(n) - n\log(e) + \frac{1}{2}\log(2\pi) + \frac{1}{2}\log(n)$$
$$= O\left(n\log(n) - n + \frac{1}{2}\log(n)\right)$$
$$= O(n\log(n))$$

שאלה 4:

ב. מחפשים את האינדקס של המערך arr איפה שנמצא בו את הערך k.
אם היינו סתם עושים binarySearch רגיל, אז הסיבוכיות הייתה שווה ל $O(\log(n))$, עם n המספרי האיברים ב-arr.
וצריכים סיבוכיות של $O(\log(m))$, עם m מספר האיברים שמופיעים לפני k במערך.
לכן יצרתי משתנה i כך שכל עוד ש $arr[2^i] < k$ אז אני מגדל $low = 2^i$ וגם $i{+}{+}$. ככה זה יקח $\log(i)$ צעדים כדי למצוא את ה low המתאים.
אז אני מכניס בתוך משתנה high את 2^{i+1} (במידה ו- $arr.length-1 > 2^{i+1}$ אז $high = arr.length-1$)
וככה יש לי סביבה של אינדקסים: אני יודע ש k נמצא בין $arr[2^i]$ ו- $arr[2^{i+1}]$
ואז אני מפעיל את binarySearch רגיל עם הערכים low ו- high הקודמים, שסיבוכיות קטן מ $\log(n)$ כי מספר האיברים בין 2^i ו- 2^{i+1} קטנה ממש מ n.
לכן קיבלנו: $O(\log(i) + \log(i) + \log(i)) = O(\log(i))$, ובגלל ש $m \geq i$
$\Leftarrow O(\log(m))$.