Assignment 1

CMPE 300, Analysis of Algorithms, Fall 2022

Bahadır Gezer - 2020400039

November 2022

Answer 1

A.

c_1 and c_2 is chosen where $c_1, c_2 > 0$ for some $n > n_0$. So $n \to \infty$. Definition of Θ is used.

$$
\begin{array}{rcll}
 & f(n) \in \Theta(n^5 log(n)) & & \text{– Equality of } f(n) \\
 & n^4 log(n^8 n!) + 4n^3 \in \Theta(n^5 log(n)) & & \text{– Definition of } \Theta \\
\lim\limits_{n\to\infty} c_1 n^5 logn \leq & \lim\limits_{n\to\infty} (n^4 log(n^8 n!) + 4n^3) & \leq \lim\limits_{n\to\infty} c_2 n^5 logn & \text{– Divide by } n^5 logn \\
\lim\limits_{n\to\infty} c_1 \leq & \lim\limits_{n\to\infty} \dfrac{n^4 log(n^8 n!) + 4n^3}{n^5 logn} & \leq \lim\limits_{n\to\infty} c_2 & \text{– Sum law \& evaluate limit} \\
c_1 \leq & \lim\limits_{n\to\infty} \dfrac{n^4 log(n^8 n!)}{n^5 logn} + \lim\limits_{n\to\infty} \dfrac{4n^3}{n^5 logn} & \leq c_2 & \text{– Simplify } n\text{'s} \\
c_1 \leq & \lim\limits_{n\to\infty} \dfrac{log(n^8 n!)}{nlogn} + \lim\limits_{n\to\infty} \dfrac{4}{n^2 logn} & \leq c_2 & \text{– Evaluate limit} \\
c_1 \leq & \lim\limits_{n\to\infty} \dfrac{log(n^8 n!)}{nlogn} + 0 & \leq c_2 & \text{– Stirling's formula} \\
c_1 \leq & \lim\limits_{n\to\infty} \dfrac{log\left(n^8 \sqrt{2\pi n} \left(\frac{n}{e}\right)^n\right)}{nlogn} & \leq c_2 & \text{– L'Hôpital's rule} \\
c_1 \leq & \lim\limits_{n\to\infty} \dfrac{logn + \frac{17}{2n}}{logn + 1} & \leq c_2 & \text{– L'Hôpital's rule} \\
c_1 \leq & \lim\limits_{n\to\infty} \dfrac{\frac{1}{n} - \frac{17}{2n^2}}{\frac{1}{n}} & \leq c_2 & \text{– Multiply by } \frac{n}{n} \\
c_1 \leq & \lim\limits_{n\to\infty} \left(1 - \dfrac{17}{2n}\right) & \leq c_2 & \text{– Sum rule} \\
c_1 \leq & \lim\limits_{n\to\infty} 1 - \lim\limits_{n\to\infty} \dfrac{17}{2n} & \leq c_2 & \text{– Evaluate limit} \\
c_1 \leq & 1 - \lim\limits_{n\to\infty} \dfrac{17}{2n} & \leq c_2 & \text{– Evaluate limit} \\
c_1 \leq & 1 - 0 & \leq c_2 & \\
c_1 \leq & 1 & \leq c_2 &
\end{array}
$$

Real values for c_1 and c_2 can be chosen from the resulting constraints. Case A is true.

B.

c is chosen where $c > 0$ for some $n > n_0$. So $n \to \infty$. Definition of Ω is used.

$$f(n) \in \Omega(n^5\sqrt{n}) \qquad \text{–Equality of } f(n)$$

$$n^4 log(n^8 n!) + 4n^3 \in \Omega(n^5\sqrt{n}) \qquad \text{–Definition of } \Omega$$

$$\lim_{n\to\infty} cn^5\sqrt{n} \le \lim_{n\to\infty} n^4 log(n^8 n!) + 4n^3 \qquad \text{–Divide by } n^5\sqrt{n}$$

$$\lim_{n\to\infty} c \le \lim_{n\to\infty} \frac{n^4 log(n^8 n!) + 4n^3}{n^5\sqrt{n}} \qquad \text{–Sum rule \& evaluate limit}$$

$$c \le \lim_{n\to\infty} \frac{n^4 log(n^8 n!)}{n^5\sqrt{n}} + \lim_{n\to\infty} \frac{4n^3}{n^5\sqrt{n}} \qquad \text{–Simplify } n\text{'s}$$

$$c \le \lim_{n\to\infty} \frac{log(n^8 n!)}{n\sqrt{n}} + \lim_{n\to\infty} \frac{4}{n^2\sqrt{n}} \qquad \text{–Evaluate limit}$$

$$c \le \lim_{n\to\infty} \frac{log(n^8 n!)}{n\sqrt{n}} + 0 \qquad \text{–Stirling's formula}$$

$$c \le \lim_{n\to\infty} \frac{log\left(n^8\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)}{n\sqrt{n}} \qquad \text{–L'Hôpital's rule}$$

$$c \le \lim_{n\to\infty} \frac{logn + \frac{17}{2n}}{\frac{3\sqrt{n}}{2}} \qquad \text{–L'Hôpital's rule}$$

$$c \le \lim_{n\to\infty} \frac{\frac{1}{n} - \frac{17}{2n^2}}{\frac{3}{4\sqrt{n}}} \qquad \text{–Sum rule \& simplify } n\text{'s}$$

$$c \le \lim_{n\to\infty} \frac{4}{3\sqrt{n}} - \lim_{n\to\infty} \frac{34}{3n^{3/2}} \qquad \text{–Evaluate limit}$$

$$c \le \lim_{n\to\infty} \frac{4}{3\sqrt{n}} - 0 \qquad \text{–Evaluate limit}$$

$$c \le 0$$

Real values of c cannot be chosen from the resulting constraints. Case B is false.

C.

c is chosen where $c > 0$ for some $n > n_0$. So $n \to \infty$. Definition of ω is used.

$$f(n) \in \omega(n^4) \qquad \text{–Equality of } f(n)$$

$$n^4 log(n^8 n!) + 4n^3 \in \omega(n^4) \qquad \text{–Definition of } \omega$$

$$\lim_{n\to\infty} cn^4 \le \lim_{n\to\infty} n^4 log(n^8 n!) + 4n^3 \qquad \text{–Divide by } n^4$$

$$\lim_{n\to\infty} c \le \lim_{n\to\infty} \frac{n^4 log(n^8 n!) + 4n^3}{n^4} \qquad \text{–Sum rule}$$

$$c \le \lim_{n\to\infty} \frac{n^4 log(n^8 n!)}{n^4} + \lim_{n\to\infty} \frac{4n^3}{n^4} \qquad \text{–Simplify } n\text{'s}$$

$$c \le \lim_{n\to\infty} log(n^8 n!) + \lim_{n\to\infty} \frac{4}{n} \qquad \text{–Evaluate limit}$$

$$c \leq \lim_{n\to\infty} log(n^8 n!) + 0 \qquad -\text{Stirling's formula}$$

$$c \leq \lim_{n\to\infty} log\left(n^8\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right) \qquad -\text{Product rule}$$

$$c \leq \lim_{n\to\infty} log\left(n^8\sqrt{2\pi n}\right) + \lim_{n\to\infty} log\left(\left(\frac{n}{e}\right)^n\right) \qquad -\lim_{n\to\infty}\frac{n}{e} = \infty, \text{ so } \lim_{n\to\infty}\infty^n = \infty$$

$$c \leq \lim_{n\to\infty} log\left(n^8\sqrt{2\pi n}\right) + \infty \qquad -\text{Evaluate limit}$$

$$c \leq \infty + \infty$$

$$c \leq \infty$$

Real values of c can be chosen from the resulting constraints. Case C is true.

D.

c is chosen where $c > 0$ for some $n > n_0$. So $n \to \infty$. Definition of $\mathcal{O}$ is used.

$$f(n) \in \mathcal{O}(n^4 logn) \qquad -\text{Equality of } f(n)$$

$$n^4 log(n^8 n!) + 4n^3 \in \mathcal{O}(n^4 logn) \qquad -\text{Definition of } \mathcal{O}$$

$$\lim_{n\to\infty} n^4 log(n^8 n!) + 4n^3 \leq \lim_{n\to\infty} cn^4 logn \qquad -\text{Divide by } n^4 logn$$

$$\lim_{n\to\infty} \frac{n^4 log(n^8 n!) + 4n^3}{n^4 logn} \leq \lim_{n\to\infty} c \qquad -\text{Sum rule}$$

$$\lim_{n\to\infty} \frac{n^4 log(n^8 n!)}{n^4 logn} + \lim_{n\to\infty} \frac{4n^3}{n^4 logn} \leq c \qquad -\text{Simplify } n\text{'s}$$

$$\lim_{n\to\infty} \frac{log(n^8 n!)}{logn} + \lim_{n\to\infty} \frac{4}{nlogn} \leq c \qquad -\text{Evaluate limit}$$

$$\lim_{n\to\infty} \frac{log(n^8 n!)}{logn} + 0 \leq c \qquad -\text{Stirling's formula}$$

$$\lim_{n\to\infty} \frac{log\left(n^8\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)}{logn} \leq c \qquad -\text{L'Hôpital's rule}$$

$$\lim_{n\to\infty} \frac{logn + \frac{17}{2n}}{\frac{1}{n}} \leq c \qquad -\text{Rearrange } 1/n$$

$$\lim_{n\to\infty} nlogn + \frac{17}{2} \leq c \qquad -\text{Evaluate limit}$$

$$\infty \leq c$$

Real values of c cannot be chosen from the resulting constraints. Case D is false.

Answer 2

Require: n is a positive odd integer
Require: k is a uniformly distributed random integer between 1 and n

1: **function** $anonymous(k, n)$
2: $x \leftarrow 1$
3: $arr \leftarrow initialize\_random\_array(n)$
4: **if** $k = \lceil n/2 \rceil$ **then**
5:     **for** $i \leftarrow 0$ **to** $n - 1$ **do**
6:         find_all_subsets(arr)
7:     **end for**
8: **else if** $k < n/2$ **then**
9:     **for** $j \leftarrow 0$ **to** $n - 1$ **do**
10:         **for** $k \leftarrow n - 1$ **to** 1 **by** $k = k/2$ **do**
11:             $x \leftarrow x * 2$
12:         **end for**
13:     **end for**
14: **else**
15:     **for** $i \leftarrow 0$ **to** $n - 1$ **do**
16:         $x \leftarrow x + 1$
17:     **end for**
18: **end if**
19: **end**

for loop 1 - line 5: This loop will iterate **from** $\mathbf{0}$ **to** $\mathbf{n-1}$, which is $\mathbf{n}$ iterations. Each iteration will call **find_all_subsets(arr)**. **find_all_subsets** finds all subsets of the array **arr**, and each subset counts as one basic operation. To make analysis simpler, elements in **arr** is assumed to be unique, this effectively makes **arr** a mathematical set. The number of subsets for a set is $\mathbf{2^y}$ where $\mathbf{y}$ is the size of the set. Thus, each iteration of the loop will have $\mathbf{2^n}$ basic operations. So the complexity comes out to be $\mathbf{f_{5-7}(n,k) = n \cdot 2^n}$

for loop 2 - line 10: This loop will iterate **from** $\mathbf{n-1}$ *-which is an even integer-* **to** $\mathbf{1}$ by halving at each iteration. For the sake of simplicity, lets consider $\mathbf{n-1}$ as $\mathbf{2^z}$ where $\mathbf{z}$ is some positive integer. It will be generalized using interpolation later.

Iteration 1 **-** $k = 2^z$
Iteration 2 **-** $k = 2^{z-1}$
Iteration 3 **-** $k = 2^{z-2}$
$\vdots \qquad \vdots$
Iteration m **-** $k = 2^{z-m-1} = 1$

As seen above, the loop will iterate $\mathbf{m}$ times. Solving the equation $\mathbf{2^{z-m-1} = 1}$ for $\mathbf{m}$ we get $\mathbf{m = z - 1}$. Substituting $\mathbf{z = log(n-1)}$ into the equation we get $\mathbf{m = log(n-1) - 1}$. Thus, the loop will have $\mathbf{log(n-1) - 1}$ iterations. Each iteration has $\mathbf{1}$ basic operation. So, the complexity comes out to be $\mathbf{f_{10-12}(n,k) = log(n-1) - 1}$.

The final complexity is only valid for $\mathbf{n = 2^z + 1}$. This function can be squeezed its way into the $\mathbf{\Theta(nlogn)}$ complexity class. **nlogn** is both Θ-invariant under scaling and eventually non-decreasing. $\mathbf{f_{10-12}(n,k)}$ is eventually non-decreasing. These are enough to interpolate the complexity function so that it is valid for all natural number values of $\mathbf{n}$.

for loop 3 - line 9: This loop will iterate **from** $\mathbf{0}$ **to** $\mathbf{n-1}$, which is $\mathbf{n}$ iterations. Each iteration will run the program from lines 10 to 12. That part of the program has a complexity of $\mathbf{f_{10-12}(n,k) = log(n-1) - 1}$. Thus, each iteration will have $\mathbf{log(n-1) - 1}$ basic operations. So the complexity comes out to be $\mathbf{f_{9-13}(n,k) = n \cdot log(n-1) - n}$

for loop 4 - line 15: This loop will iterate **from** $\mathbf{0}$ **to** $\mathbf{n-1}$, which is $\mathbf{n}$ iterations. Each iteration has $\mathbf{1}$ basic operation. So the complexity comes out to be $\mathbf{f_{15-17}(n,k) = n}$

The *anonymous* function has three different conditional branches to enter based on the value $\mathbf{k}$. It does not have any more operations to execute after the termination of each branch. So the complexity of the function depends on which branch it enters.

conditional branch 1 - line 4: This branch just executes **for loop 1** and exits. So the complexity of this branch is the same as the complexity of **for loop 1**, which is $\mathbf{n \cdot 2^n}$

conditional branch 2 - line 8: This branch has two nested for loops, then it exits. In this case since the outer for loop takes the inner for loops complexity into account, just considering the complexity of the outer for loop is enough. So the complexity of this branch comes out to be $\mathbf{n \cdot \log(n-1) - n}$.

conditional branch 3 - line 14: This branch just executes **for loop 4** and exits. So the complexity of this branch is the same as the complexity of **for loop 4**, which is $\mathbf{n}$.

Worst Case

The worst case behaviour of this function happens when $\mathbf{k} = \lceil \mathbf{n/2} \rceil$. Then the function will enter the branch with the highest complexity, which is $\mathbf{n \cdot 2^n}$.

$$
\begin{aligned}
W(n,k) &= n \cdot 2^n \leq n \cdot 2^n \\
&\implies W(n,k) \in \mathcal{O}(n2^n) \\
W(n,k) &= n \cdot 2^n \geq n \cdot 2^n \\
&\implies W(n,k) \in \Omega(n2^n) \\
W(n,k) \in \Omega(n2^n) \land W(n,k) \in \mathcal{O}(n2^n) &\implies \mathbf{W(n,k) \in \Theta(n2^n)}
\end{aligned}
$$

□

Average Case

Assume that the probability of $\mathbf{k} = \lceil \mathbf{n/2} \rceil$ is $\boldsymbol{\rho}$. $\mathbf{k}$ can have $\mathbf{n}$ different equally likely values and $\boldsymbol{\rho}$ happens for only one value of $\mathbf{k}$. Thus, $\boldsymbol{\rho} = \mathbf{\frac{1}{n}}$. Since $\mathbf{n}$ is an odd integer, $\lceil \mathbf{n-1} \rceil$ value equally divides the range of possible $\mathbf{k}$ values in half - $k = \lceil n/2 \rceil$ *is omitted from the two halves.* So $\mathbf{k}$ has equal probability of ending up in each half, which is $\mathbf{\frac{1-\boldsymbol{\rho}}{2}}$ or $\mathbf{\frac{n-1}{2n}}$.

This makes the set $\mathbf{T_{n,k}}$ have 3 elements. Which each mapping out to one of the three conditional branches the function might enter.

I_1 - $\rho(I_1) = \frac{1}{n}$, $\tau(I_1) = n \cdot 2^n$

I_2 - $\rho(I_2) = \frac{n-1}{2n}$, $\tau(I_2) = n \cdot log(n-1) - n$

I_3 - $\rho(I_3) = \frac{n-1}{2n}$, $\tau(I_3) = n$

$$
\begin{aligned}
A(n,k) = &\sum_{I \in T_{n,k}} \tau(I) \cdot \rho(I) \\
&= (n \cdot 2^n) \cdot \frac{1}{n} + (n \cdot log(n-1) - n) \cdot \left(\frac{n-1}{2n}\right) + n \cdot \left(\frac{n-1}{2n}\right) \\
&= 2^n + \frac{1}{2} \cdot n \cdot log(n-1) - \frac{1}{2} \cdot log(n-1)
\end{aligned}
$$

Some positive constants $\mathbf{c_1}$, $\mathbf{c_2}$ *and* $\mathbf{n_0}$ *can be found such that* $\mathbf{A(n,k)}$ *is squeezed by the function above when* $\mathbf{n > n_0}$.

$$\implies A(n,k) \in \Theta(2^n + \frac{1}{2} \cdot n \cdot log(n-1) - \frac{1}{2} \cdot log(n-1))$$

Terms of lower order -$\mathbf{n \cdot \log(n)}$ *and* $\mathbf{\log(n)}$*- can be omitted from the complexity class.*

$$\implies \mathbf{A(n,k) \in \Theta(2^n)}$$

□