2 Answer I

(2) i = 1 — 1
Sum = 57 — 1
finished = False — 1
while ((i < = n) and (not finished))
  for j = 1 to i do
    if (COND(i))
      Sum = Sum + (i / j)
      for K = 1 to log(n) do
        Sum = Sum + K
    else
      Sum = Sum - (j / i)
  if COND (Sum) then
    finished = true
    i = 57
  else
    i = i + 1

Sum = Sum / i + 63

$2+4+(\log(n)-1) \times 3+2$

$(\log(n)-1)(1+2)+2$

1

$$(i-1)(8+3(\log(n)-1)+2)+2 = \sum_{i=0}^{i=n-1} \left((i-1)(8+3(\log(n)-1)+2)+2\right)$$

Answer

inside outer while

let $X = (i-1)(8+3(\log(n)-1)+2)+2$

$T(n) = n(X+1)+1+4$

Further explanation

Here the total number of operations performed will be sum of $(i-1)$ over all values from 0 to $(n-1)$

→ now these values are coming from outer while loop.

A Assumption COND(i) → true COND(Sum) = False to get worst case time complexity

Q2 Answer II

(1) This means we can write

$$T(n) = n(X+1)+10 = \sum_{i=0}^{n-1} \left((i-1)(8+3(\log(n)-1)+2)+2\right)$$
$+4$

$$= ((8+3(\log(n)-1))+2) \sum_{i=0}^{n-1} (i-1) + 2\sum_{i=0}^{n-1} 1$$

$$T(n) = ((8+3(\log(n)-1))+2)\left(\frac{n(n-1)}{2}\right) + 2n + 4$$

~~and The~~ here $T(n)$ is worst case time complexity

and asymptotic worst-case will be

$$O(n^2 \log(n))$$

(3) Answer I

~~Inner~~ inner for Loop

For k=1 to n do

n+13+2 if COND (sum)

sum= sum / (k * i)+j

1 ← while ((j <= n) and (not finished)) do

O(1) finished = true.

this will run in Constant time so it I'm Gonna Consider it as 1 step O(1)

↓

Because while loop will only execute when finished = False But as soon as while loop will execute for first time finish will become true and loop will stop executing so Basically 2 steps only, But I'm Gonna write it just 1 step for simplicity -> it doesn't really matter.

$$\to (n-1)(n+16+2)+2 = n^2+16n+2n-n-16-2+2$$

$$n^2+17n-16$$

Considering outer loop

$$(n\log(n)-1)(n^2+17n-\cancel{14}^{-12}+2)+2$$

$$n^3\log n - n^2 + 17n^2\log(n) - 17n - 12n\log(n) + 12 + 2$$

I'm Going to use limit method as nothing is mentioned about the way we are suppose to use.

3rd Answer II

As per the limits definition

$$f(n) = O(g(n)) \Rightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)}$$

$$T(n) = n^3 \log(n) - n^2 + 17n^2 \log(n) - 17n - 12^n \log(n) + 14$$

The simple function that will Bound the Running time is $g(n) = n^3 \log(n)$

$$\lim_{n\to\infty} \frac{T(n)}{g(n)} = 1$$

It means Growth rate of $T(n)$ is proportional to $g(n)$ as n grow towards infinity.

$n^3\log(n)$

So the time Complexity of $T(n)$ is $O(n^3 \log(n))$.

Other way

→ Here the term with highest degree of n is $n^3 \log(n)$ so it will dominate the Running time of ALGORITHM as $n\to\infty$

Here lower degree terms Can Be ignored as they have smaller impact on time Complexity

So time Complexity of function is

$O(n^3 \log(n))$.

3 Answer III

A. In the above question I haven't considered "$O(N) \to (n+1)$" as a constant factor as no information is given whether n is constant or not.

But if n is constant then we can ignore it and time complexity will be $O(n^2 * \log(n))$.

What I have considered that n will depend on input.

4th Answer I

(ii) Considering the inner loop

for i=1 to n*n do] $(n^2-1)(T(P_1)+T(P_2)+1+2)+2$
$T(P_1)+T(P_2)+1$ [$y = x / (P_2(n) + P_1(n))$

if ($P_3(n)$)
  if ($P_4(n)$)] $3+T(P_4)+T(P_3)$
    tsum = tsum + tsum
else
  for i=1 to log(n) do] $(\log(n)-1)(0+2)+2$
  if ($P_4(n)$)
    $y = y * i - j$] $T(P_4)+3$
  tsum = tsum / y

$y = P_1(n)$
sum = sum - x+5] $2+T(P_1)$

- Here we will have two Cases:-

Case 1: if ($P_3(n)$) // is true
$3+T(P_4)+T(P_3)$ [if ($P_4(n)$)
    tsum = tsum + tsum.

$(n^2-1)(T(P_1)+T(P_2)+3)+2+3+T(P_4)+T(P_3)+T(P_4)+3$
$+2+T(P_1)$

$\Rightarrow 10+2T(P_4)+T(P_3)+T(P_1)+n^2T(P_1)+n^2T(P_2)+n^2*3$
$-T(P_1)-T(P_2)-3$

$\Rightarrow 7+2T(P_4)+T(P_3)+n^2(T(P_1)+T(P_2)+3)-T(P_2)$

After outer For Loop:-

$(n-1)(7+2T(P_4)+T(P_3)+n^2(T(P_1)+T(P_2)+3)-T(P_2)$
$+2)+2$

4th Answer II

$9n + 2nT(P_4) + nT(P_3) + n^3(T(P_1) + T(P_2)+3)$
$\quad -T(P_2))\times n$
$-9 - 2T(P_4) - T(P_3) - n^2(T(P_1) + T(P_2)+3)$
$\quad + T(P_2) + 5$

$= n(9 + 2T(P_4) + T(P_3) - T(P_2))$
$\quad + n^3(T(P_1) + T(P_2) + 3) - n^2(T(P_1)+T(P_2)+3)$
$\quad - 4 - 2T(P_4) - T(P_3) + T(P_2)$

Parameterized time Complexity for Case I.

Case 2: if $(P_3(n))$ // False

else
for $i=1$ to $\log(n)$ do $2(\log(n-1))2+2$
$2\log(n) - 2+2$
$= 2\log(n)$.

$1+T(P_3) + 2\log(n)$

$(n^2-1)(T(P_1) + T(P_2) + 3) + 2 + 1 + T(P_3) + 2\log(n)$
$\quad + T(P_4) + 3 + 2 + T(P_1)$

$= n^2(T(P_1) + T(P_2)+3) - T(P_1) - T(P_2) - \cancel{3} + \cancel{3} + T(P_3) + 2\log(n)$
$\quad + T(P_4) + 5 + T(P_1)$
$= n^2T(P_1) + n^2T(P_2) + 3n^2 - T(P_2) + T(P_3) + 2\log(n)$
$\quad + T(P_4) + 5.$

After outer for loop:-

$(n-1)(T(P_1)(n^2) + T(P_2)(n^2-1) + T(P_3)(1) +$
$\quad T(P_4)(1) + 3n^2 + 2\log(n) + 5 + 2)$
$\quad + 2 + 3$

$\Rightarrow T(P_1)(n^3) + T(P_2)(n^3-n) + T(P_3)(n) +$
$\quad T(P_4)(n) + 3n^3 + 2n\log n + 7n)$
$- T(P_1)(n^2) - T(P_2)(n^2-1) - T(P_3)(1) - T(P_4) - 3n^2$
$\quad - 2\log(n) - 7) + 5$

$-25 + 40 - 12$ $\quad C \times 16$ $\quad 4 \times 4 \le 16 - 16 + 12 \le 5 \times 4$ $\quad 16 \le 12 \le 20$ $\quad 1 - 8 + 12$ $\quad 4 \le 5 \le 6$ $\quad 16 - 32 + 12$ $\quad 12$ $\quad 28$ $\quad 32$ $\quad (-4)$

(5)

(a) $f(n) = (n-2)(n-6)$ is not $\Theta(n^2)$.

$= n^2 - 6n - 2n + 12$

$= n^2 - 8n + 12$

- As per definition we have to demonstrate that there are two positive constants C_1 and C_2 and an positive initial value n_0 : $n \ge n_0$

We need to prove $\quad \therefore g(n) = n^2$

$$C_1 g(n) \le f(n) \le C_2 g(n)$$

To show that $f(n)$ is $\Theta(n^2)$

$$C_1 n^2 \le (n^2 - 8n + 12) \le C_2 n^2 \quad \therefore n \ge n_0$$

Taking C_1 as a

After simplifying: $(1 - C_2) * n^2 + 8n - 12 < 0$

As $n \to \infty$ Left hand side expression become positive.

Therefore, $f(n)$ is not $O(n^2)$

$C_1 n^2 \le (n^2 - 8n + 12)$

$(1 - C_1) * n^2 - 8n + 12 \ge 0$

As $n \to \infty$ L.H.S become negative

so $f(n)$ is not $\Omega(n^2)$

Therefore, we can conclude

$f(n)$ is not $\Theta(n^2)$

(3) $f(n) = n^d + 10n^2 \quad d \geq 2 \quad O(n^d)$

We need to show $f(n) = n^d + 10n^2 <= C n^d$
for all $n >= n_0$ for some Constants C, n_0.

I[st] Case when $d = 2$

$$f(n) = n^2 + 10n^2 <= Cn^2$$
$$11n^2 <= Cn^2$$
$$11 <= C \qquad \underline{C = 11} \qquad n \geq 1$$
$$\underline{n_0 = 1}$$

II[nd] Case $d > 2$

$$f(n) = n^d + 10n^2 <= Cn^d$$
$$10n^2 <= n^d(C-1)$$
$$10 <= n^{d-2}(C-1)$$

Here $d > 2$ means $d = 2, 3, 4, 5, 6$
let $X = d-2 = 0, 1, 2, 3, 4$ ---

So logically $n^{d-2} = n^X = n^0$ or n^1 or n^2 ---

Considering $n \geq 1$
$n^{d-2} = n^X$ will Be Positive

So we Can Take $C = 11$
for condition to Be always true.

II[nd] Case
Therefore $f(n)$ is $O(n^d)$ for $C = 11$, $n_0 = 1$

I[st] Case $f(n)$ is $O(n^2)$ for $C = 11$, $n_0 = 1$
↳ Here $d = 2$ so $O(n^2)$ and $O(n^d)$ are Same

(c) $f(n) = 10^{127} 2^n$
?

$$10^{127} 2^n >= C 3^n$$
$$10^{127} \left(\frac{2}{3}\right)^n >= C$$

Here as $n \to \infty$ $10^{127} \left(\frac{2}{3}\right)^n \to 0$

(1)

(c) By defination we have to show

$$10^{127} 2^n >= C 3^n \text{ for all } n >= n_0$$
for some constants C, n_0

$$10^{127} \left(\frac{2}{3}\right)^n >= C$$

Here n approaches ∞ $10^{127} \left(\frac{2}{3}\right)^n \to 0$

So $10^{127} \left(\frac{2}{3}\right)^n >= C$ can not be true left hand side asymptotically goes to 0 when $n \to \infty$ hence $C > 0$ cann ot greater then any constant for sufficiently large value of n.

		0	1	2	3	4	5	6
Question 6		y_i	T	A	T	C	G	A
0	x_i	0	0	0	0	0	0	0
1	G	0	↑ 0	↑ 0	↑ 0	↑ 0	↖ 1	1
2	A	0	↑ 0	↖ 1	← 1	← 1	← 1	↖ 2
3	A	0	↑ 0	↖ 1	← 1	← 1	← 1	↖ 2
4	G	0	↑ 0	↑ 1	← 1	← 1	↖ 2	← 2
5	C	0	↑ 0	↑ 1	↑ 1	↖ 2	↑ 2	↑ 2
6	C	0	↑ 0	↑ 1	↑ 1	↖ 2	↑ 2	↑ 2
7	T	0	↖ 1	↑ 1	↖ 2	↑ 2	↑ 2	↑ 2
8	A	0	↑ 1	↖ 2	↑ 2	↑ 2	↑ 2	↖ 3

The LCS is "AGA" and I have highlighted Path By "↑" dark arrows ↳ LCS associated with this Path is "AGA" and I have attached .java file of the Code.

Path:- (x_p, y_p) (8,6) → (7,5) → (6,5) → (5,5) → (4,5) → (3,4) → (3,3) → (3,2) → (2,1) → (1,1) → (0,1)

↓

Base Case where i = 0; ~~→ will ret~~