21k-3881

Q6)

$\Rightarrow n^2 + 8n + 15 = O(n^2)$

Sol) $f(n) \leq c \times g(n)$

$n^2 + 8n + 15 \leq n^2 + 8n^2 + 15n$

$\Rightarrow n^2 + 8n + 15 \leq 9n^2 + 15n$

$\therefore c = 9, n_0 = 1$

$n \geq 1$

$T(n) = O(n^2)$

$\Rightarrow 5n^2 \log_2 n + 2n^2 = O(n^2 \log_2 n)$

Sol) $f(n) \leq c \times g(n)$

$5n^2 \log_2 n + 2n^2 \leq 5n^2 \log_2 n + 2n^2 \log_2 n$

$\therefore c = 5, n_0 = 2$

$\therefore n \geq 2$

$T(n) = O(n^2 \log_2 n)$

21k-3881

Q7)

(i) For Big O Notation (O):

This notation is ~~represented~~ represents upper bound or worst case time complexity an upper limit on how algorithms run time grows as input size increases.

(ii) Big Omega (Ω) Notation:

This notation represents lower bound or best case time complexity of an algorithm or a function. It provides a lower limit on how fast algorithm can perform for given input size.

(iii) Big Theta Notation (Θ):

This represents upper and lower bound of an algorithm's time complexity, providing a tight bound on it's performance. It describes exact behaviour of algorithm.

2K-3881

Q8)

(i) $T(n) = \sqrt{2}\, T\left(\frac{n}{2}\right) + \log n$

Sol) $T(n) = a\, T\left(\frac{n}{2}\right) + \theta\left(n^k \log^p n\right)$

$a = \sqrt{2}$, $b = 2$, $k = 0$, $P = 1$

$f(n) = \log n$

$\log_b a = \log_2 \sqrt{2} = \frac{1}{2} \log_2 2 = \frac{1}{2}$

$\therefore k = 0$

$\log_b a > k$ then $\theta\left(n^{\log_b a}\right)$

$\theta\left(n^{\log_2 \sqrt{2}}\right) = \theta\left(n^{1/2}\right)$

$\boxed{\Rightarrow \theta\left(\sqrt{n}\right)}$

(ii) $T(n) = 64\, T(n/8) - n^2 \log n$

Sol) $a = 64$, $b = 8$, $d = 2$

A/c to Master theorem:

$\theta\left(n^{\log_b a} \log^{k+1} n\right)$

$\theta\left(n^{\log_8 64} \log^{2+1} n\right)$

$\theta\left(n^{\log_8 8^2} \log^3 n\right)$

$\theta\left(n^2 \cancel{\log_8 8} \log^3 n\right)$

$\boxed{\theta\left(n^2 \log^3 n\right)}$

(iii) $T(n) = 16T(n/4) + n!$

Sol) $T(n) = 16T(n/4) + n!$

$a = 16, \ b = 4$

According to Master Theorem

$\Theta(n^{\log_b a} \log^{k+1} n)$

but $f(n) = n! \notin \Theta(n^d)$

$\therefore$ In this case $n!$ is not bounded by any polynomial, so the Master theorem does not apply.

Q9)

(i) $T(n) = 2T\left(\frac{n}{2}\right) + n\log n$

Sol:

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

$T(n) = 2 \times 2T\left(\frac{n}{4}\right) + 2\left(\frac{n}{2}\log\frac{n}{2}\right) + n\log n$

$T(n/4) = 2T(n/8) + \frac{n}{4}\log\frac{n}{4}$

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

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

21k-3881

$n\log\frac{n}{2} + \dots + n\log n$

$T(n) = 2^3 T\left(\frac{n}{2^3}\right) + n\left(\log\frac{n}{2^2} + \log\frac{n}{2^1} + \dots + \log n\right)$

$T(n) = 2^k T\left(\frac{n}{2^k}\right) + n\left(\log\frac{n}{2^{k-1}} + \log\frac{n}{2^{k-2}} + \dots + \log n\right)$

$\because \log n/2^k/2 = \log 2n/2^k$

$\therefore 2^k = n$

$k = \log n$

$T(n) = nT(n/n) + n\left(\log n/2^k/2 + \log n/2^k/2^2 + \dots + \log n\right)$

$T(n) = n + n\left(\log 2\cancel{n}/\cancel{n} + \log 4\cancel{n}/\cancel{n} + \dots + \log n\right)$

$T(n) = n + n\left(\log 2 + \log 4 + \dots + \log n\right)$

21k-3881

$$T(n) = n + n(1+2+3+\dots+\log n)$$

$$T(n) = n + n\frac{\log n(\log n + 1)}{2}$$

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

$\therefore$ Time Complexity:

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

(ii) $T(n) = 8T\left(\frac{n}{2}\right) + n^2$

Sol.

$$T\left(\frac{n}{2}\right) = 8T\left(\frac{n}{4}\right) + \frac{n^2}{4}$$

$$T(n) = 8 \times 8T\left(\frac{n}{4}\right) + \frac{8n^2}{4} + n^2$$

$$T\left(\frac{n}{4}\right) = 8T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$T(n) = 8 \times 8 \times 8T\left(\frac{n}{8}\right) + \frac{64n^2}{16} + \frac{8n^2}{4} + n^2$$

$$T\left(\frac{n}{8}\right) = 8T\left(\frac{n}{16}\right) + \frac{n^2}{64}$$

$$T(n) = 8 \times 8 \times 8 \times 8T\left(\frac{n}{16}\right) +$$

21k-3881

$$\frac{512n^2}{64} + \frac{464n^2}{16} + \frac{8n^2}{4} + n^2$$

$$T(n) = 8^4\, T(n/2^4) + n^2(8+4+2+1)$$

$$T(n) = 8^k\, T\left(n/2^k\right) + n^2\left(2^{k-1} + 2^{k-2} + 2^{k-3} + 2^{k-4} + \dots + 2^2 + 2^1 + 2^0\right)$$

$$T(n) = 8^k\, T\left(n/2^k\right) + n^2\left(2^0 + 2^1 + 2^2 + \dots + 2^{k-4} + 2^{k-3} + 2^{k-2} + 2^{k-1}\right)$$

$\therefore n = 2^k$

$\therefore \log n = k \log_2 2$

$$T(n) = n^3\, T\left(2^k/2^k\right) \cancel{+ n^2(1+2}$$

$$+ n^2\left(\frac{1(2^{k-1+1} + 1)}{2-1}\right)$$

$$T(n) = n^3(1) + n^2(2^k + 1)$$

$$T(n) = n^3 + n^2(n+1)$$

$$T(n) = n^3 + n^3 + n^2$$

$$T(n) = 2n^3 + n^2$$

$\therefore$ Time Complexity:

$$O(n^3)$$

21k-3881

Q10)

(i) For all positive (+ve) $f(n)$:

$\omega(f(n)) + O(f(n)) = \theta(f(n))$

$\Rightarrow \omega(f(n)) = f(n)$

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

$f(n) = n^2$

$g(n) = n$

$\Rightarrow n^2 + n = \theta(n^2)$

Hence true for all positive values.

(ii) For all positive $f(n)$:

$f(n) + O(f(n)) = \theta(f(n))$

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

$$= \frac{\cancel{n}}{\cancel{n}}$$

$\Rightarrow 1 \neq \infty$

Hence it is false for all positive values.

(iii) For all positive $f(n)$, $g(n)$ and $h(n)$;

if $n = O(g(n))$ and $f(n) = \Omega(h(n))$

then $g(n) + h(n) = \Omega(f(n))$

Eg) $f(n) = n$ $g(n) = n^2$ and $h(n) = 1$

$f(n) = O(g(n))$ and $f(n) = \Omega(h(n))$

Let $g(n) + h(n) \Rightarrow n^2 + 1 \Rightarrow \Omega(n)$

Hence it is true for all value.