Date.
Page No.

① Asymptation notation are the methematical natations used to describe the running time of an algorithm correspondit to the input ~~not~~ trends the particular value on a limiting ~~value~~.

~~Ex~~ → Types of notation

1) Θ Notation :- It is the way to expres the both the lower bound and the upper bound of an algorithm's running time

2) ~~Big~~ O Notation — It is the formal way to expres the upper band of an algorithm's running time.

3) Omega Natation Ω - It is the formal way to expres the lower bound of an algorithm's running time.

② $\sum_{i=1 (i=i \times 2)}^{n}$ $1+1+1 \cdots k$ times

$2^k \geq = n$

$2^k = n$

taking log both side

$k \log 2 = \log n$

$k \log 2 = \log n$

$k = \frac{\log n}{\log 2}$

$k = \log_2 n$

Ans $O(\log n)$

(3) $T(n) = \begin{cases} 3T(n-1) & n>0 \\ 1 & n=0 \end{cases}$

$T(n) = 3T(n-1)$ —(1)

let $n = n-1$

putting n in eq (1)

$T(n-1) = 3T(n-2)$ —(3)

let $n = n-2$

putting n in eq (1)

$T(n) = 3T(n-2)$ —(4)

putting (4) in (3)

$T(n) = 3^3 T(n-3)$

$T(n) = 3^k T(n-k)$

let $n-k=0$

$n=k$

$T(n) = 3^n T(0)$

$= O(3^n)$

(1) (5) $i = 1, 2, 3, 4, 5, 6 \ldots$

sum of $= 1+3+6+10+15+21$ —— (1)

also $= 1+3+6+10 \ldots T_{n-1} + T_2$ —(2)

(1) $= 1+2+3+4+ \ldots n = T_2$

$T_k = 1+2+3+4+\ldots k$

$T_{k^2} = \frac{1}{2}(k)(k+1)$

for k iteration

$1+2+3 \ldots k <= n$

$\frac{k(k+1)}{2} \le n$

$O(k^2) <= n$

$k = O(\sqrt{n})$

$T(n)$ $(O\sqrt{n})$

$i^2 = n$

$i = \sqrt{n}$

$i = 1\ 2\ 3\ 4 \ldots \sqrt{n}$

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

$$T(n) = O(n)$$

for $k = k \cdot 2$

$k = 1, 2, 4, 8 \ldots n$

$GP = 1, 2, \cancel{3}4, 8 \ldots n$

Sum of n-terms $= \dfrac{a(r^n - \cancel{1})}{r-1}$

$$n = \frac{1(2^k - 1)}{2-1} = 1\frac{(2^k - 1)}{2}$$

$$n = 2^k - 1$$

taking log on both side

$\log_2 n = k \log_2 2 - \log_2 1$

$\log_2 n = k$

i	j	k
1		
2	log n	log n * log n
|	log n	
|	|	
n	log n	

$\Rightarrow O(\log n * \log n) \Rightarrow O(n \log^2 n)$.

fun(int n)
{ }

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

$a = 1,\ b = 3$ $\qquad f(n) = n^2$

$c = \log_3 1 = 0$

$\Rightarrow n^c = 1 > f(n)$

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

(9) for $i = n \Rightarrow i = 1, 2, 3, 4 ---- n = n$
for $i = 2 \Rightarrow j = 1, 3, 5 ---- n = n/2$
for $i = 3 \Rightarrow 1, 4, 7 ------ n \quad n/3$

for $i = n \Rightarrow f = 1$

$$\sum_{j=n}^{1} n + \frac{n}{2} + \frac{n}{3} ---- 1$$

$$\sum_{j=n}^{1} n\left[1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} -- \frac{1}{n}\right]$$

$$\sum_{j=n} n \log n)$$

$$\Rightarrow T(n) = n \log n$$

$$T(n) = O(n \log n)$$

(10) as given n^k & c^n
relation b/w n^k & c^n is
$n^k = O(c^n)$
as $n^k \leq ac^n$
$\forall n \geq n_0$ as constant $a > 0$
for $n_0 = 1$
$c = 2$
$\Rightarrow 1^k \leq a2$
$\Rightarrow n_0 = 1$ & $c = 2$