Nitin Singh
'I, 21

Design and Analysis of Algorithm

Tutorial - 1

Ans1:- Asymptotic Notation:- It means towards infinity. They are used to tell the complexity of an algorithm having input size very Large.
It is priory analysis.

* Different types of asymptotic notations are

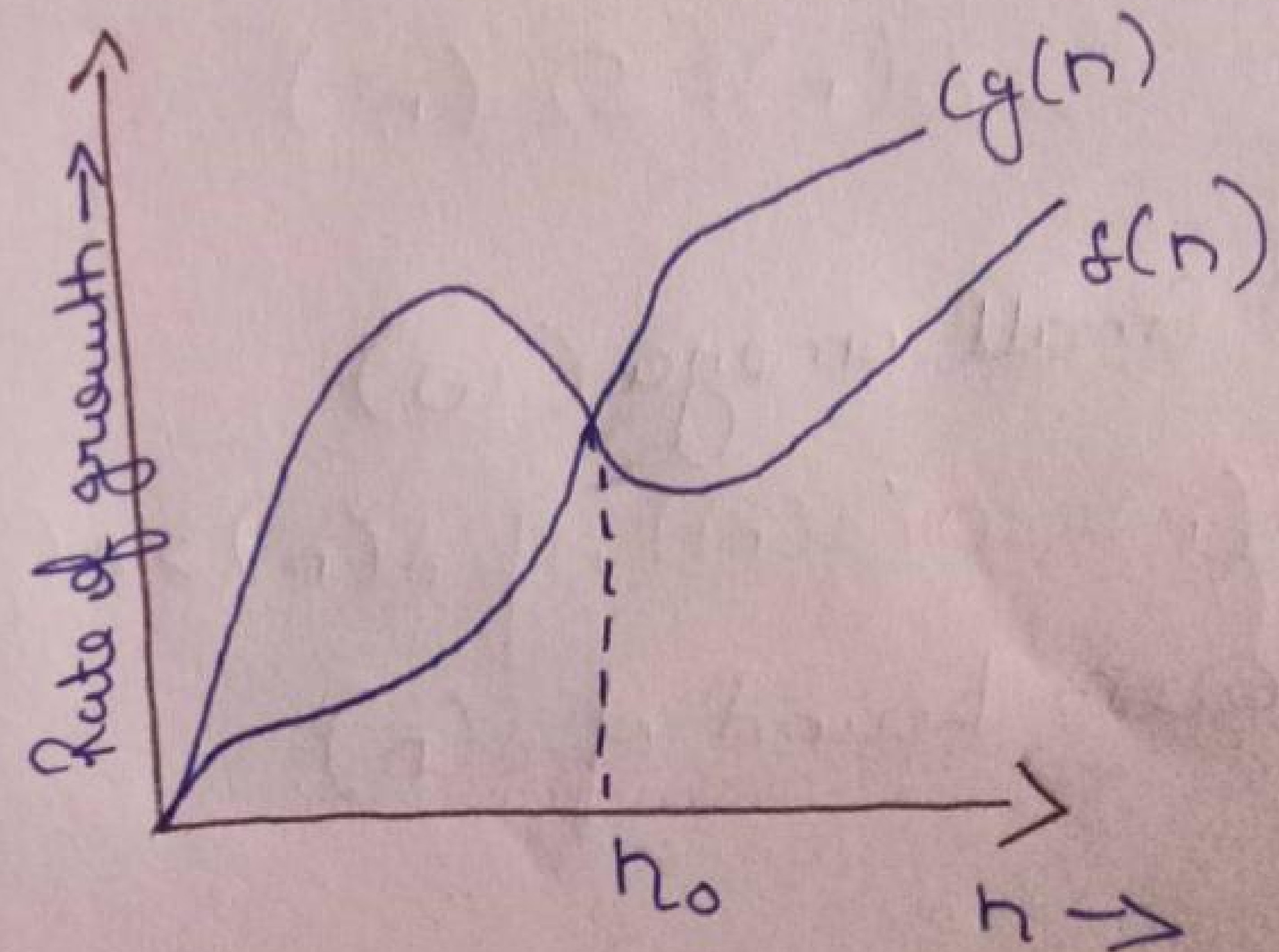
(i) Big oh Notation

$f(n) = O(g(n))$, if $0 \le f(n) \le c(g(n))$ $\forall n \ge n_0$ g(n) is tight upper bound of f(n)

Example

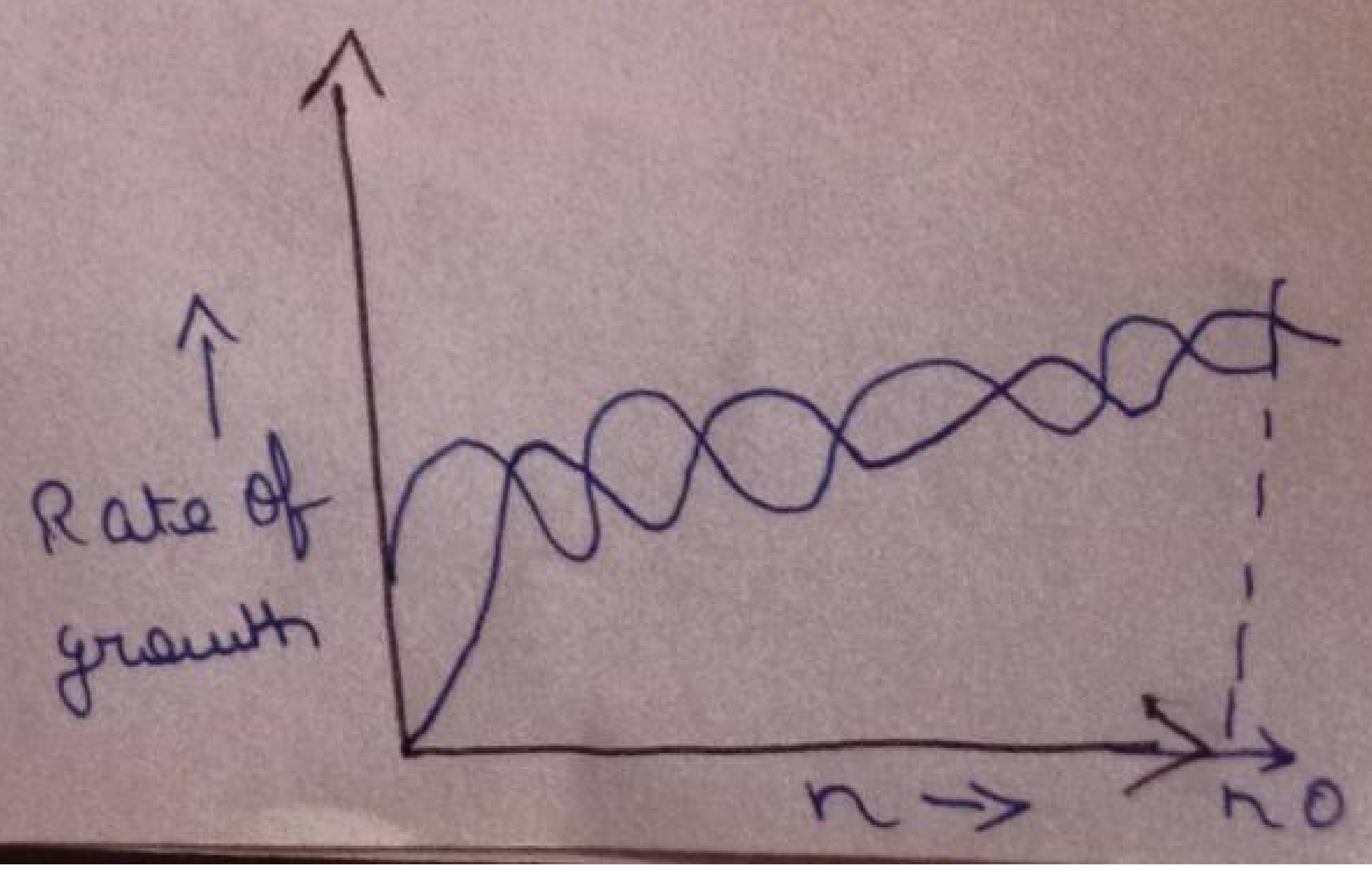
```
for (int i=0 ; i<n ; i++)
{
    cout<< i<< endl;
}
```

$T(n) = O(n)$



(ii) Small oh Notation

$f(n) = o(g(n))$, if $f(n) < c(g(n))$ $\forall n > n_0$ & $\forall c > 0$
g(n) is upper bound of f(n).



(iii) Big Omega (Ω)

$f(n) = \Omega(g(n))$, if $f(n) \geq c(g(n))$ $\forall$ $n \geq n_0$ & some constant $c>0$

$g(n)$ is tight lower bound of $f(n)$.

* Example :- $f(n) = 6n^2 + n + 1$, $g(n) = n^2$

$$0 \leq cg(n) \leq f(n)$$

$$0 \leq c \cdot n^2 \leq 6n^2 + n + 1$$

$$c \leq 6 + \frac{1}{n} + \frac{1}{n^2}$$ on putting $n = \infty$, $\frac{1}{n} = \frac{1}{\infty} = 0$

$$c \leq 6$$

$$6n^2 \leq 6n^2 + n + 1 \Rightarrow (n = 1)$$

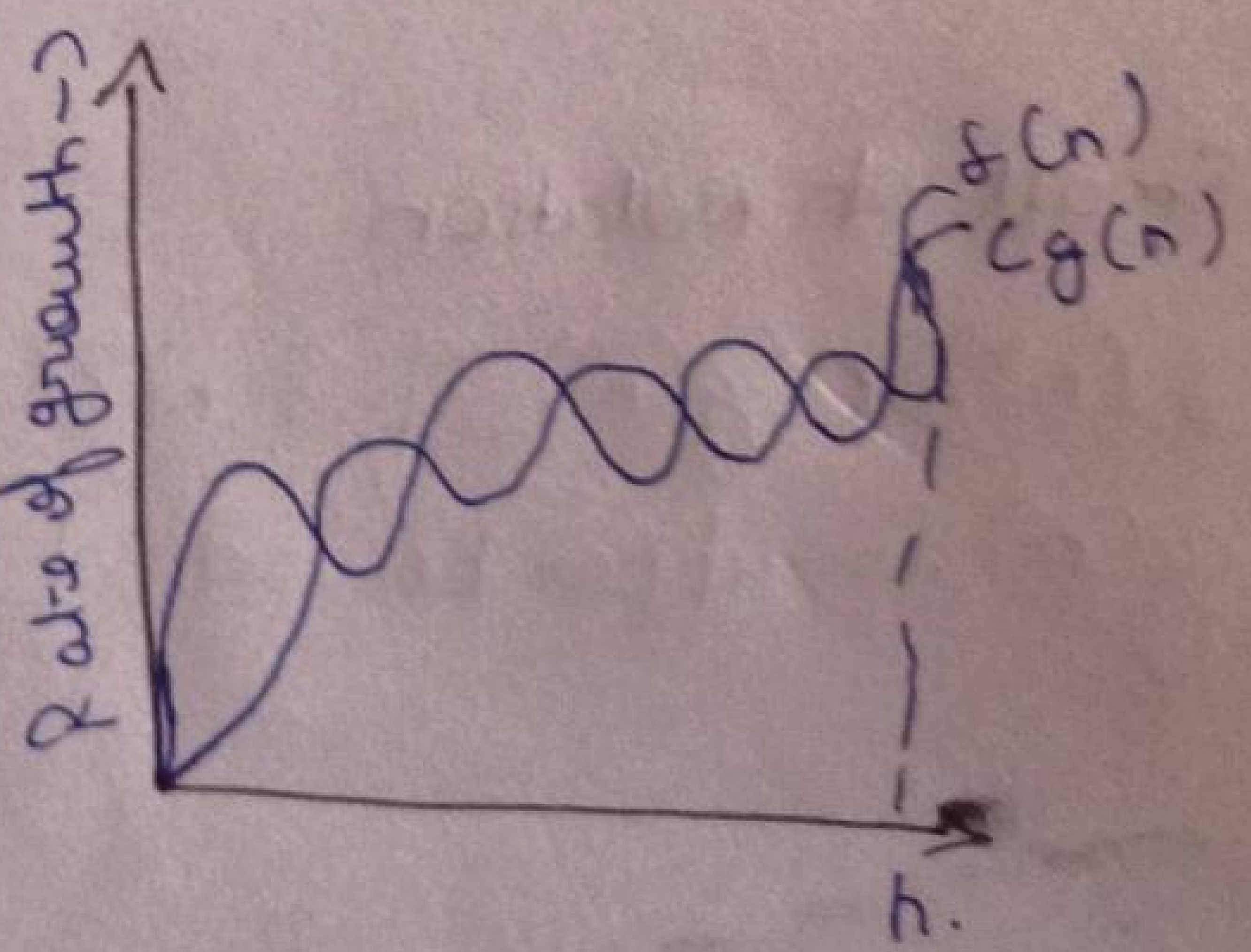
$6 \leq 6 + 1 + 1 \Rightarrow 6 \leq 8$ True $\quad \because c>0$ and $n \geq n_0$ $(n = 1)$

$n_0 = 1$

$$f(n) = \Omega(n^2)$$

(iv) Small omega (ω)

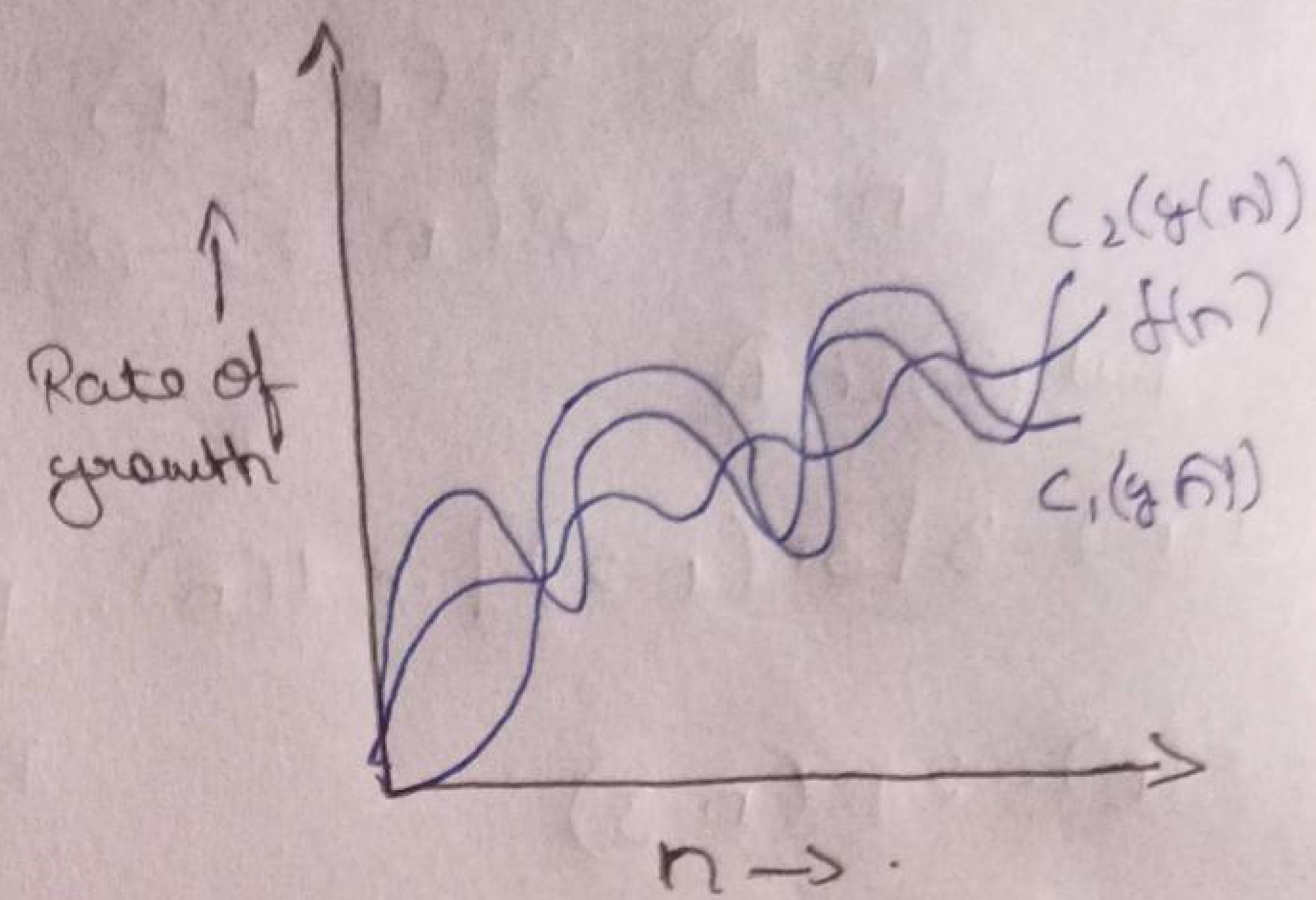
- $f(n) = \omega(g(n))$, if $f(n) > c(g(n))$ $\forall$ $n > n_0$ & $\forall$ $c>0$ $g(n)$ is the lower bound of $f(n)$.



(v) Theta (θ)

$f(n) = \theta(g(n))$, if $c_1(g(n)) \leq f(n) \leq c_2(g(n))$

$\forall\ n \geq \max(n_1, n_2)$ and some constant c_1 & $c_2 \geq 0$



Ans 2 $(p=1, \text{ to } n)\{p = p*2\}$

i would have 1, 2, 4, 8, 16, - - - - n

let say these are k terms.

It is a G.P with $a=1$, $r=2$

Now, k^{th} term $= t_k = ar^{k-1}$

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

$$n = 2^{k-1}$$

Taking $\log_2$ on both sides

$\log_2 n = \log_2(2^{k-1})$

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

$\log n = (k-1) \Rightarrow k = 1 + \log_2 n$

$T(n) = O(k) = O(1 + \log n) \Rightarrow O(\log n)$

Q3 $T(n) = 3T(n-1)$ — (1)

by backward substitution

$\because T(n) = 3T(n-1)$

$T(n-1) = 3T(n-1-1)$

$T(n-1) = 3T(n-2)$ — (2)

Put (2) in (1)

$T(n) = 3[3T(n-2)] \Rightarrow T(n) = 9T(n-2)$ — (3)

$T(n-2) = 3T(n-3)$

$T(n) = 27T(n-3)$

continue for k times.

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

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

assume $n-k = 0 \Rightarrow n = k$

$T(n) = 3^k T(0)$. $\quad (\because T(0) = 1)$

$T(n) = 3^k$

$\boxed{T(n) = O(3^n)}$

Scanned by TapScanner

Q4 $T(n) = \{2T(n-1) - 1$, if $n>0$, otherwise $1\}$.

$T(n) = 2T(n-1) - 1$ — (1)

by using Back substitution method.

$T(n) = 2[2T(n-2)-1]-1$ $\quad$ $T(n) = 2T(n-1)-1$

$= 2^2 T(n-2) - 2 - 1$ — (2) $\quad$ $T(n-1) = 2T(n-2) - 1$

$= 2^2[2T(n-3)-1]-2-1$ $\quad$ $T(n-2) = 2T(n-3) - 1$

$= 2^3 T(n-3) - 4 - 2 - 1$

continue for k times.

$T(n) = 2^k[(n-k) - 2^{k-1} - 2^{k-2} - - - - -]$

Assume $n-k = 0 \Rightarrow n = k$

$= 2^n T(0) - 2^{n-1} - 2^{n-2} - - - - 1$

$= 2^n - 2^{n-1} - 2^{n-2} - - - - 1.$

$= 2^n - [\underbrace{2^{n-1} + 2^{n-2} - - - - + 1}_{\text{G.P k terms}}]$

$a = 2^{n-1}$, $r = 2^{-1} \Rightarrow \frac{1}{2}$

sum of G.P

$$= \frac{a(1-r^{n-1})}{1-r}$$

$$= 2^{n-1}\frac{\left(1 - \left(\frac{1}{2}\right)^{n-1}\right)}{1-\frac{1}{2}} = 2^{n-1}\frac{\left(1 - \frac{(1/2)^n}{1/2}\right)}{1/2}$$

Scanned by TapScanner

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

$= \cancel{2^n} \frac{(2^n - 2)}{\cancel{2^n}} \Rightarrow 2^n - 2$

$= 2^n - [2^n - 2] \Rightarrow 2 \Rightarrow T(n) = O(2)$

$T(n) = O(1)$ Ans

Ans5 $S = 1, 3, 6, 10, 15, ---n$

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

$k^2 = 2n$

$k = \sqrt{n}$

$O(\sqrt{n})$

$\because$ t_{k-1} would be constant.

Ans6 $1^2, 2^2, 3^2, ----n$

Let say k terms

$t_k = k^2$

$n = k^2 \Rightarrow k = \sqrt{n}$

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

$p * p$

$1 \times 1 \Rightarrow 1^2$

$2 \times 2 = 2^2$

$3 \times 3 = 3^2$

$k \times k = k^2 = n$

Scanned by TapScanner

Ans 7 $i = \frac{n}{2}, \frac{n}{2}+1, \frac{n}{2}+2 ------ n$

$= \frac{n}{2}, \frac{n+2}{2}, \frac{n+4}{2} ------ n$

General form $= \frac{n+0*2}{2} + \frac{n+1*2}{2} + \frac{n+2*2}{2}, ---n$

$= \frac{n+k*2}{2} \quad (k = 0,1,2,----n)$

Total terms $= k+1$

$+ k+1 = n.$

$= \frac{n+(k+1)*2}{2} = n \Rightarrow 2n = n+(k+1)*2$

$n-2 = 2k.$

$k = \frac{n}{2} - 1$

i	j	k
$\frac{n}{2}$	logn times	$(\log n)^2$
$\frac{n+2}{2}$	logn times	$(\log n)^2$
⋮		
n	logn times	$(\log n)^2$

$(\frac{n}{2}-1)$ times

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

$= \frac{n}{2}\log^2 n - \log^2 n \quad \Rightarrow \boxed{T(n) = O(n\log^2 n)}$

Scanned by TapScanner

Ans 8 Function call would be n, n-3, n-6, n-9 ---- 1
let say k terms.

AP, $a = n$, $d = -3$,

$a_n = a + (n-1)d$.

$1 = n + (k-1)(-3)$.

$1 = n - 3k + 3$

$3k = n + 2$

$k = \frac{n+2}{3}$

$\because$ Function have recursive calls $\frac{n+2}{2}$ times

Time complexity for two inner loop $= n^2$

$$\left(\frac{n+2}{2}\right) \cdot n^2 \Rightarrow n^3$$

$$\boxed{T(n) = O(n^3)}$$

Ans 9 for i.e. (outer loop).
when $i = 1 \rightarrow j = 1, 2, 3, 4 --- n \Rightarrow n$
when $i = 2 \rightarrow j = 1, 3, 5, 7 --- n \Rightarrow \frac{n}{2}$
when $i = 3 \rightarrow j = 1, 4, 7 --- n \Rightarrow \frac{n}{3}$

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

$$\sum_{j=n}^{1} n\left(1 + \frac{1}{2} + \frac{1}{3} + --- + \frac{1}{n}\right).$$

$$\boxed{O(n \log n)}$$ Ans

Ans 10 As given n^k and c^n.

relation b/w n^k and c^n is $n^k = O(c^n)$

as $n^k \leq a c^n \; \forall \; n \geq n_0$ for a constant $a > 0$

for $n_0 = 1$

$c = 2$

$1^k \leq a 2^1$

$\therefore n_0 = 1$ and $c = 2$.