Date.....................

Expt. No..................... Tutorial-1 Page No.....................

Design and Analysis of Algorithms

Karan Agarwal D-12
2016805

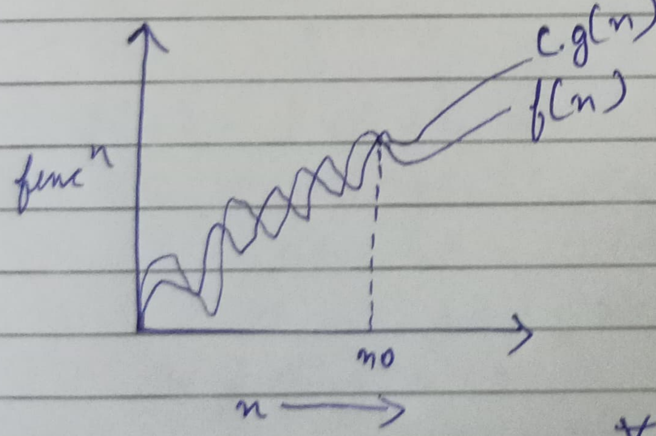
Ans 1:-

Asymptotic Notations:- These notations are used to tell the complexity of an algorithm, when input is very large. These are mathematical notations used to describe running time of an algorithm when the input tends towards a particular value or a limiting value.

• Different Asymptotic Notations:-

(i)- Big-Oh (O):-

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



g(n) is 'tight' upper bound.

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

iff

$f(n) \leq c.g(n)$

$\forall$ $n \geq n_0$ and some constant, $c > 0$

Teacher's Signature.....................

eg :-

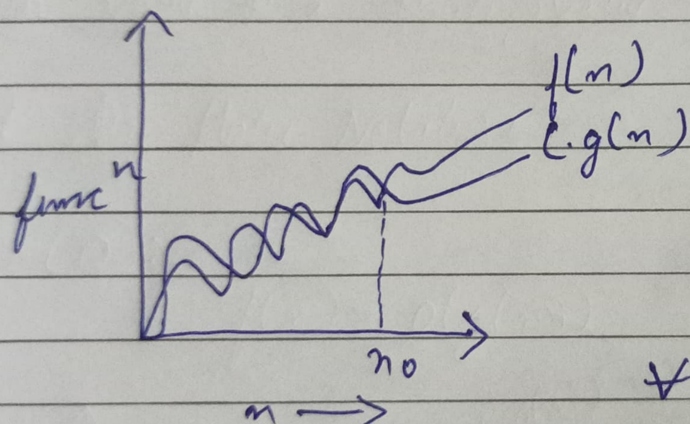
```
for(i=1; i<=n; i++)
{
    print(i);       —— O(1)
}
```

$= T(n) = \Theta(n)$

i
1
2
3
⋮
n times $\Rightarrow O(n)$

iii) Big Omega (Ω):-

$$f(n) = \Omega(g(n))$$



$g(n)$ is 'tight' lower bound.

$f(n) = \Omega(g(n))$ iff $f(n) \geq c.g(n)$ $\forall$ $n \geq n_0$, and some constant, $c > 0$.

Eg. $f(n) = 2n^2 + 3n + 5$ $\quad g(n) = n^2$

$\Rightarrow \because 0 \leq c.g(n) \leq f(n)$

$\because 0 \leq c.n^2 \leq 2n^2 + 3n + 5$

$\Rightarrow c \leq 2 + 3/n + 5/n^2$

On putting $n = \infty$, $\quad \Rightarrow \frac{3}{n} \to \infty, \frac{5}{n^2} \to \infty$

$\Rightarrow c = 2,$

$2n^2 \leq 2n^2 + 3n + 5$

On putting $n = 1$,

$2 \leq 2 + 3 + 5$ $\quad 2 \leq 10$ True

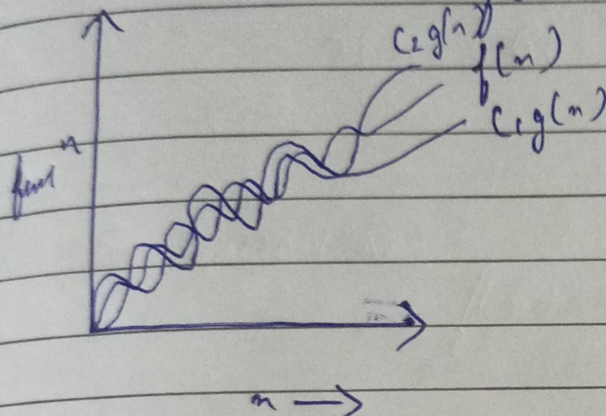
 $\Rightarrow c = 2, \; n = n_0 = 1$

$$0 \le 2n^2 \le 2n^2 + 3n + 5$$

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

iii:- Big - Theta (Θ):-

$$f(n) = \Theta(g(n))$$



$g(n)$ is both "tight" upper and lower bound of $f(n)$

$f(n) = \Theta(g(n))$

iff

$$c_1 . g(n) \le f(n) \le c_2 . g(n)$$

$\forall\ n \ge \max(n_1, n_2)$

and some constant

$c_1 > 0$, $c_2 > 0$.

Eg:- $f(n) = 10 \log_2 n + 4$, $g(n) = \log_2 n$

$$= f(n) \le c_2 g(n)$$

$$= 10 \log_2 n + 4 \le 10 \log n + \log n.$$

$$= 10 \log_2 n + 4 \le 11 \log_2 n$$

$$c_2 = 11$$

$$4 \le 11 \log_2 n - 10 \log n$$

$$4 \le \log_2 n$$

$$16 \le n$$

Here, $\forall\ n \ge 16$

$n_2 = 16$

and $c_2 = 11$

Date..................

Expt. No.................... Page No....................

$$f(n) \geq c_1 g(n)$$

$$10 \log n + 4 \geq 2 \log n$$

$c_1 = 1$, $n > 0$

$\Rightarrow n_1 = 1$ $\Rightarrow n = max(n_1, n_2) = n = 16$

$$\log_2 n \leq 10 \log_2 n + 4 \leq 11 \log_2 n$$

$c_1 = 1$, $c_2 = 11$

$\Rightarrow \Theta(\log_2 n)$

(iv) Small Oh (o):-

$$f(n) = o(g(n))$$

$g(n)$ is the upper bound of the funⁿ $f(n)$.

$$f(n) = o(g(n))$$

when, $f(n) \leq c \cdot g(n)$

$\forall n > n_0$

and $\forall$ constant, $c > 0$.

(v):- Small Omega (ω):-

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

$g(n)$ is lower bound of the funⁿ $f(n)$

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

when

$$f(n) > c \cdot g(n)$$

$\forall n > n_0$

and $\forall c > 0$.

Teacher's Signature..........................

Date........................

Expt. No...................................... Page No......................................

Ans 2. for ($i = 1$ to n) { $i = i * 2$; }

$\Rightarrow i = \underbrace{1, 2, 4, 8, 16 \; --- \; n}_{k \text{ terms}}$

$a_0 = 1 \;,\; r = 2$

k^{th} term

$t_k = ar^{k-1}$

$n = 1.2^{k-1}$

$n = 2^{k-1}$

taking log on both sides,

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

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

$\log_2 n = k - 1$

$\Rightarrow k = 1 + \log_2 n \qquad [\because \log_a a = 1]$

$\Rightarrow T(n) = O(k)$

$= O(\cancel{1} + \log_2 n)$

$= O(\log n)$.

3- $T(n) = \{3T(n-1)$ if $n > 0$, otherwise $1\}$

$\Rightarrow$

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

put $n = n-1$ in eq ①

$T(n-1) = 3T(n-2) \quad - ②$

put this in eq ①

Teacher's Signature..

$T(n) = 3[3T(n-2)]$ — ③

put $n = n-2$ in Eq ①,

$T(n-2) = 3T(n-3)$ — ④

put this value in Eq ③

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

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

Generalised form:-

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

put $n-k=0$

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

put $T(0) = 1$

$\Rightarrow T(n) = 3^n$

$\Rightarrow O(3^n)$.

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

$\Rightarrow$

$T(n) = 2T(n-1) - 1$ — ①

put $n-1$ in Eq ①

$\Rightarrow T(n-1) = 2T(n-2) - 1$ — ②

put this value in Eq ①

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

$T(n) = 4T(n-2) - 2 - 1$ — ③

put $n = n-2$ in Eq ①

$\Rightarrow T(n-2) = 2T(n-3) - 1$ — ④

put this value in eq ③,

$T(n) = 4[2T(n-3) - 1] - 2 - 1$

Date.....................

Expt. No.. Page No....................................

$T(n) = 8\,T(n-3) - 4 - 2 - 1$

Generalised form :-

$T(n) = 2^n\,T(n-k) - 2^{k-1} - 2^{k-2} - \cdots - 1$

put $n - k = 0$

$n = k$, $T(0) = 1$ (given)

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

$= 2^n - 2^{n-1} - 2^{n-2} - \cdots - 1$

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

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

$\Rightarrow$ Sum of GP $= \dfrac{2^{n-1}[1 - (1/2)^{n-1}]}{1 - 1/2} = 2^n - 2$

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

$= O(2) = O(1)$.

5-

```
int i=1, s=1;
while (s<=n) {
    i++; s = s+i;
    printf("#"); ->O(1)
}
```

s
1
3
6
10
15
;
n th
k times

$\Rightarrow S = \underbrace{1, 3, 6, 10, 15 - - - n}_{K \text{ terms}}$

$\Rightarrow$ Kth term,

$t_k = t_{k-1} + k$

$k = t_n - t_{n-1}$ — ①

Teacher's Signature..............................

Date.......................
Expt. No.......................... Page No.........................

$\Rightarrow K = n - t_{k-1}$

$\Rightarrow$ loop means k times

$\Rightarrow TC = O(1+1+1+ n - t_{k-1})$

but, $t_{n-1} = c$ (constant)

$\Rightarrow T.C = O(3+n-2))$

$= O(n)$.

6:- Time Complexity of -

```
void func (int n) {         — O(1)
int i, count = 0;           — O(1)
for(i = 1; i*i ≤ n; i++)
     count++;               — O(1)
}
```

$i*i$
1^2
2^2
3^2
4^2
|
|
|
n

$\Rightarrow i*i \Rightarrow \underbrace{1^2, 2^2, 3^2, 4^2, 5^2 - - n}_{k \text{ terms}}$

$\Rightarrow$ K, n terms :-

$t_k = K^2$

$= K^2 = n$

$K = n^{1/2}$

$\Rightarrow TC = O(\cancel{1} + \cancel{1} + \cancel{1} + n^{1/2} + \cancel{1})$

$= O(n^{1/2}) = O(\sqrt{n})$.

Teacher's Signature.........................

7 - Time Complexity of -

```
void function (int n) {          —— O(1)
  int i, j, k, count = 0;        —— O(1)
  for (i = n/2; i <= n; i++)
    for (j = 1; j <= n; j = j * 2)   log(n) times
      for (k = 1; k <= n; k = k * 2) — log(n) times
        count++;                 — O(1)
}
```

8 - Time Complexity of -

```
function (int n) {
  if (n == 1) return;      —— O(1)
  for (i = 1 to n) {       —— O(n)
    for (j = 1 to n) {     —— O(n)
      printf("*");         —— O(1)
    }
  }
  function (n-3);
}
```

for func^n call,

$\underbrace{n, n-3, n-6, n-9 ------ 1}_{k \text{ terms}}$

$\Rightarrow$ AP with $d = -3$

$\Rightarrow l = a + (k-1)d$

Date...................

Expt. No.................................. Page No...........................

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

$$\frac{1-n}{(-3)} = k-1$$

$$\Rightarrow k-1 = \frac{n-1}{3}$$

$$k = \frac{n-1+3}{3}$$

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

$\Rightarrow$ funcⁿ gives a recursive call $n+2/3$ times

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

$$= O(n^3)$$

$$i = n/2,\ n+2/2,\ n+4/2,\ \frac{n+6}{2} \text{ --- upto } n$$

$$\Rightarrow \frac{n+0\times2}{2}, \frac{n+1\times2}{2}, \frac{n+2\times2}{2}, \frac{n+3\times2}{2} \text{ -- upto } n$$

$\Rightarrow$ General form:-

$$t_k = \frac{n+k*2}{2}$$

total terms $= k+1$

$$\Rightarrow t_{k+1} = n$$

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

$$= n + 2k + 2 = 2n$$

Teacher's Signature...............................

Date..........

Expt. No.......... Page No..........

$$2K = n - 2$$

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

$\Rightarrow$

i	j	K
$n/2$	$\log n$	$(\log n)^2$
$n+2/2$	$\log n$	$(\log n)^2$
$n+4/6$	$\log n$	$(\log n)^2$
|	|	|
n	$\log n$ times	$(\log n)^2$
$(n/2 - 1)$ times		

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

$$= O\left(n/2 \log^n n - \log n\right)$$

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

9 $\Rightarrow$ TC of

```
void function(int n)
{
    for(i=1 to n)
    {
        for(j=1; j<=n; j++)
        print("*");
    }
}
```

for i $= 1 \rightarrow j = 1, 2, 3, 4, \ldots, n = n$

for i $= 2 \rightarrow j = 1, 3, 5, \ldots, n = n/2$

for i $= 3 \rightarrow j = 1, 4, 7, \ldots, n = n/3$

⋮

Teacher's Signature..........

Date..................

Expt. No.................................. Page No..................................

for $i = n \Rightarrow j = 1, \ldots, n = 1$

$$\Rightarrow \sum_{i=n}^{1} n + n/2 + n/3 + n/4 + \cdots + 1$$

$$\Rightarrow \sum_{i=n}^{1} n\left[1 + 1/2 + 1/3 + \cdots + 1/n\right]$$

$$\Rightarrow \sum_{i=n}^{1} n \log n$$

$$\Rightarrow t(n) = [n \log n]$$

$$\boxed{t(n) = O(n \log n)}$$

10 → As given, n^k & c^n

relation b/w n^k & c^n is $\boxed{n^k = O(c^n)}$

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

for $n_0 = 1$

$c = 2$

$$\Rightarrow 1^k \leq a 2^1$$

$\therefore$ $\boxed{n_0 = 1 \text{ and } c = 2}$

Teacher's Signature..................................