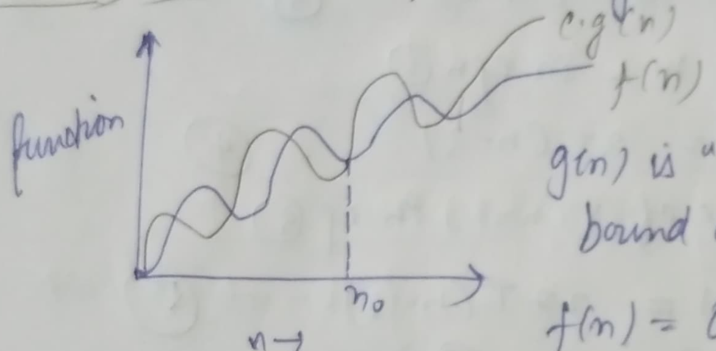
Tutorial-1 Adesh Sengar
Sec D, 14

Qus.1 what do you mean by Asymptotic notations. Define different type of notation along with e.g

Ans Asymptotic Notations: means tending to infinity, they are used to tell the complexity when i/p is very large.

→ Different types of Asymptotic Notations.

1. Big oh (O) Notation: $f(n) = O(g(n))$



g(n) is "tight" upper bound of f(n)

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

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

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

e.g
```
for (i=1; i<=n; i++)
{   print ("*");  — O(1)
}
```
$\Rightarrow T(n) = O(n)$

Q.2 what should be the time complexity of
for (i=1 to n) { i = i*2 }

values of $i = 1, 2, 8, 16, \ldots n$ — k-terms

G.P, $a=1, r=2$

kth term $t_k = ar^{k-1}$

$n = 1 \cdot 2^{k-1}$ $\quad n = 2^{k-1}$

$n = 2^{k-1}$

$\log n = (k-1)$

$k = \log_2 n + 1$

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

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

$T(n) = 3T(n-1)$ --- ①

Put $n = n-1$ in eq ①

$T(n-1) = 3T(n-2)$ --- ②

Put value of $T(n-1)$ from eq ② in eqn ①

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

$= 9T(n-2)$ --- ③

Put $n = n-2$ in eqn ①

$T(n-2) = 3T(n-3)$ --- ④

Put value of $T(n-2)$ in eq ③

$T(n) = 27T(n-3)$ --- ⑤

on generalising eq ⑤

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

Put $n-k = 0$

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

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

$\therefore T(n) = O(3^n)$

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

$T(n) = 2T(n-1) - 1$ --- ①

Put $n = n-1$ in eq ①

$T(n-1) = 2T(n-2) - 1$ --- ②

Put the value of $T(n-1)$ from eq ② in eq ①

$T(n) = 4T(n-2) - 2 - 1$ --- ③

Put $n = n-2$ in eq ①

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

Put value of $T(n-2)$ in eq ③

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

On generalising

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

Put $n-k=0$, $\Rightarrow n=k$, $T(0)=1$ (given)

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

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

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

Sum of GP = $$\frac{2^{n-1}\left[1 - \left(\frac{1}{2}\right)^{n-1}\right]}{1 - \frac{1}{2}}$$

$$= 2^n - 2$$

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

$O(2)$

$$\boxed{T(n) = O(1)}$$

⑤ what should be the t.c of

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

i	s
1	1
2	3
3	6
:	10
:	:
n	n

$\rightarrow$ k-times

$S = 1, 3, 6, \cancel{1}0, 15, \ldots n$

$\underbrace{\qquad\qquad\qquad}_{k \text{ terms}}$

k^{th} term, $t_k = t_{k-1} + k$

$k = t_k - t_{k-1}$ ---- (1)

$k = n - t_{k-1}$

loop runs k-times

$T.C = O(1+1+1+n+ \ldots t_{n-1})$

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

$\therefore TC = O(3+n-c)$

$= O(n)$

Que:(7) T.C of void function (int n)

```
{ int i, j, k, count=0;
for (i=n/2 ; i<=n ; i++)
  for (j=1 ; j<=n ; j=j*2)
    for (k=1 ; k<=n ; k=k*2)
      count++
}
```

$i \rightarrow \frac{n}{2}, \frac{n+2}{2}, \frac{n+4}{2}, \frac{n+6}{2}, \ldots$ upton

$= \frac{n+0\times2}{2}, \frac{n+1\times2}{2} + \frac{n+2\times2}{2}, \ldots n$

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

total terms = $k+1$

$t_{k+1} = n$

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

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

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

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

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

$$\Rightarrow O\left[\frac{n}{2}\log^2 n - \log n\right]$$

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

(6) T.C of void function (int n)

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

Sln:

$i \times i$

1^2

2^2

3^2

4^2

⋮

n

$$i \times i = \underbrace{1^2, 2^2, 3^2, 4^2, \ldots n}_{k\text{-terms}}$$

$$t_k = k^2$$

$$k^2 = n$$

$$k = \sqrt{n}$$

$$T.C = O(1+1+1+n^{1/2})$$

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

(8) TC 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);
}
```

Sln:- for function call

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

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

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

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

$\frac{1-n}{(-3)} = k-1 \Rightarrow k-1 = \frac{n-1}{3}$ $\boxed{k = \frac{n+2}{3}}$

Hence, function have a recursive call $\frac{n+2}{3}$ times

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

(9) T.C of void fun (int 3)

```
{  for (i=1 to n) {
       for (j=1; j<=n; j=j+i)
           printf("*");
   }
}
```

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

$i=2 \rightarrow j = 1, 3, 5, 7, \ldots n = \frac{n}{2}$

$i=3 \rightarrow j = 1, 4, 7, \ldots n = \frac{n}{3}$

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

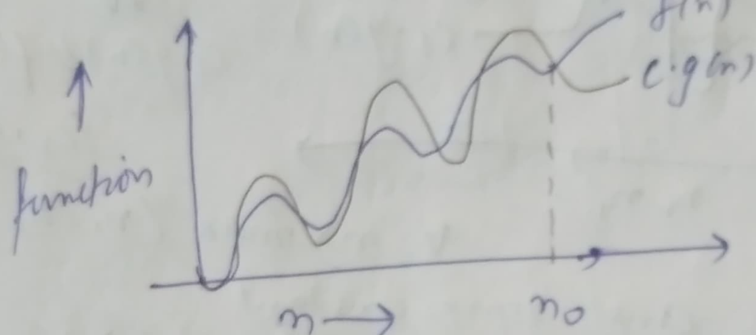
$$\sum_{j=n}^{1} n + n/2 + n/3 + \cdots + 1$$

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

$$\sum_{j=n}^{1} n \log n \qquad T(n) = O(n \log n)$$

T

2 <u>Big omega (Ω):-</u> $f(n) = \Omega(g(n))$



g(n) is "tight" lower bound of f(n)

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

iff $f(n) \geq c \cdot g(n)$

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

<u>ex.</u> $f(n) = 2n^2 + 3n + 5$, $g(n) = n^2$

$0 \leq c \cdot g(n) \leq f(n)$

$0 \leq c \cdot n^2 \leq 2n^2 + 3n + 5$

$c \leq 2 + \frac{3}{n} + \frac{5}{n^2}$

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

$\Rightarrow c = 2$

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

on putting $n = 1$

$2 \leq 2 + 3 + 5$

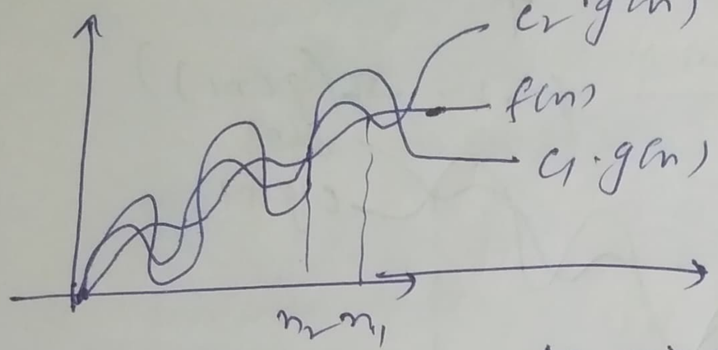
$2 \leq 10$ True

$\boxed{c = 2,\ n = n_0 = 1}$

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

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

③. <u>Big theta (Θ)</u> $f(n) = \Theta(g(n))$



$f(n) = \Theta(g(n))$
iff
$c_1 \cdot g(n) \le f(n) \le (c_2 \cdot g(n))$

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

and some constant $c_1 > 0$ & $c_2 > 0$

eg $f(n) = 10\log n + 4$, $g(n) = \log_2 n$

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

$10\log n + 4 \le 10\log n + \log n$

$10\log n + 4 \le 11\log n$

$c_2 = 11$

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

$4 \le \log_2 n$

$16 \le n$

$\forall\ n \ge 16$
$n_2 = 16$
& $c_2 = 11$

$f(n) \ge c_1 \cdot g(n)$

$10\log_2 n + 4 \ge 2\log_2 n$

$c_1 = 1,\ n > 0$

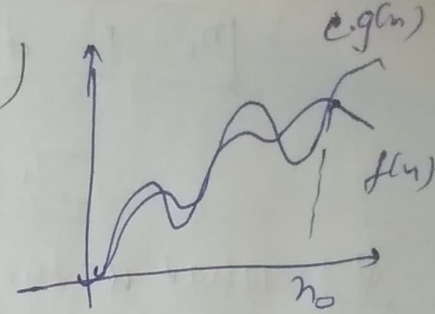
$n_1 = 1 \Rightarrow n_0 = \max(n_1, n_2) \Rightarrow n_0 = 16$

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

$c_1 = 1,\quad c_2 = 11$

$\Rightarrow \Theta(\log_2 n)$

4. Small oh (o):- $f(n) = o(g(n))$



g(n) is upper bound of f(n)

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

iff $f(n) < c \cdot g(n)$

$\forall n > n_0$ & $\forall$ constant, $c > 0$

(5) Small omega (ω):-

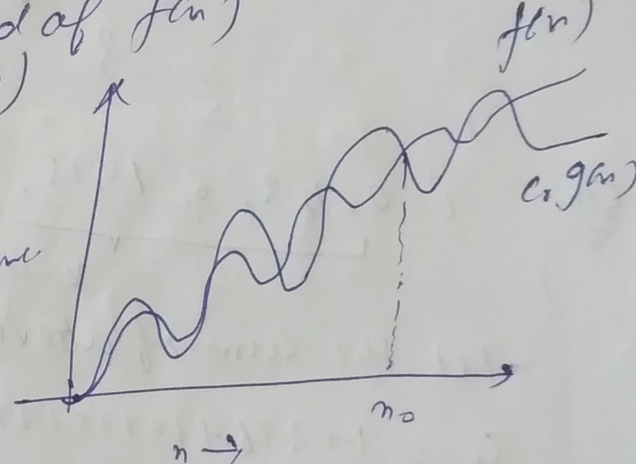
$f(n) = \omega(g(n))$

g(n) is lower bound of f(n)

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

when

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



when

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

$\forall n > n_0$

and $\forall c > 0$

(10) for the functions, n^k & c^n, what is the asymptotic notation relationship b/w these. Assume that $k >= 1$ and $c > 1$ are constants. find out the value of c and n_0. ~~for~~ for which relation holds.

As given n^k and c^n

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

as $n^k \le a c^n$ $\forall n \ge n_0$ for constant $a > 0$

for $n_0 = 1$

$c = 2$ $\quad$ $1^k \le a2^1$