Name Ayush Agarwal.
Section - I
Roll No - 30

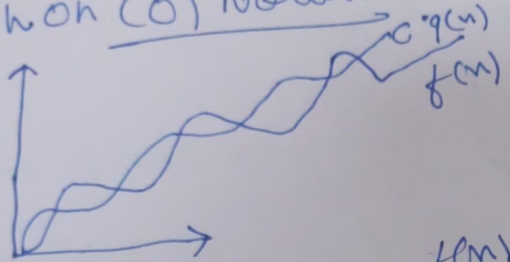
Tutorial - 01

Ques1 What do you mean by Asymptotic notations. Define different type of notation along with ex.

Asymptotic means tending to infinity. They are used to see the complexity when input is very large.

→ Different types of Asymptotic notations are:-

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



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

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

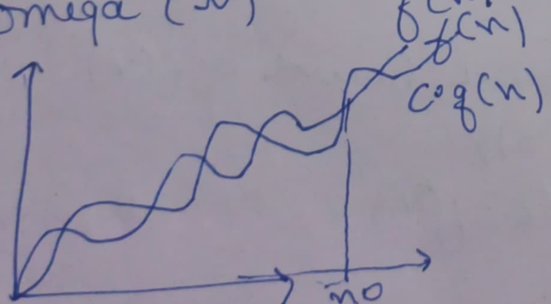
if $f(n) \le c \cdot g(n)$

$\forall n \geq n_0$ and some constant

Example. for (i=1; i<=n; i++)
{ print ("*"); → O(1)
}

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

(ii) Big omega (Ω) → $f(n) = \Omega(g(n))$



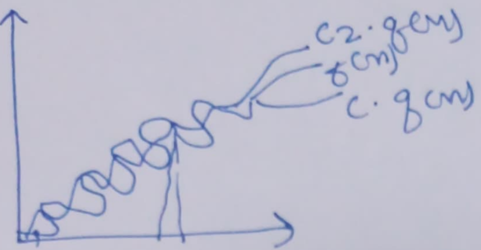
g(n) is a tight lower bound of (f(n))

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

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

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

(3) Big Theta (Θ) $f(n) = \Theta(g(n))$



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

iff

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

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

and some constant

$c_1 > 0$ & $c_2 > 0$

Ex $f(n) = 10\log_2 n + 4$, $g(n) = \log_2 n$.

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

$10 \cdot \log_2 n + 4 \le 10\log_2 n + \log_2 n$

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

$c_2 = 11$

(4) small ~~omega~~ oh (o):

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

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

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

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

$\forall\ n > n_0$ and ~~$\forall$~~ constant $c > 0$.

(5) small omega (ω) :-

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

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

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

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

$\forall\ n > n_0$

& $c > 0$.

Q What is the time complexity of

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

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

and this is a GP with $a = 1$, $r = 2$

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

$1 \cdot 2^{k-1}$

$n = 2^{k-1}$

taking log on b/s

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

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

$\log_2 n = k - 1 \Rightarrow k = 1 + \log_2 n$

Time complexity $= \underline{O(\log_2 n)}$

Q3. $T(n) = \{ 3T(n-1)$ if $n > 0$, otherwise $1 \}$.

$T(n) = 3T(n-1)$ ——— (1)

put $n = n-1$ in eqn (1)

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

$T(n-1) = 3T(n-2)$ ——— (2)

Put value of eqn (2) in eqn (1)

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

$T(n) = 9T(n-2)$ ——— (3)

Put $n = n-2$ in eqn (1)

$T(n-2) = 3T(n-3)$ ——— (4)

Put value of $T(n-2)$ in eqn (3)

$T(n) = 3(9T(n-3))$

$T(n) = 27T(n-3)$ ——— (5)

On Generalizing eqn (5)

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

Put $n - k = 0$

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

3^k $\quad$ $\underline{T(0) = 1}$

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

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

$T(n) = 2T(n-1) - 1$ —— ①

Put $n = n-1$ in eqn ①

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

$T(n-1) = 2T(n-2) - 1$ ---- ②

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

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

$T(n) = 4T(n-2) - 2 - 1$ —— ③

Put $n = n-2$ in eqn ①

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

$T(n) = 4T(n-2) - 2 - 1$ —— ③

Put $n = n-2$ in eqn ①

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

Put value of $T(n-2)$ in eqn ③

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

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

On Generating

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

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

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

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

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

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

$O(2)$

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

(5) what is the time complexity of

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

i	s
1	1
2	3
3	6
|	10
|	15
	|
	n
	k times

1, 3, 6, 10, 15 --- n

k terms

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

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

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

loop runs k times

Time complexity $= O(1+1+1+ \text{----} + n - t_{n-1})$

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

Time complexity $= O(3+n-1)$

$O(n)$

(6) Time complexity 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++; }
```

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

$\frac{n+0*2}{2} + \frac{n+1\times 2}{2} + \frac{n+2\times 2}{2}$ ---- upto n

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

total terms = k+1

$t_{k+1} = n$

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

$n+2k+2 = 2n$

$2k = n-2$

$k = n/2 - 1$

i	j	k
n/2	$\log_2 n$ times	$(\log_2 n)^2$
$\frac{n+2}{2}$	$\log_2 n$ "	"
|		
n	"	"

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

$\left(\frac{n}{2}-1\right)(\log_2 n)^2$

$O(n \log^2 n)$

Ques 6 Time Complexity of

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

i*i

1^2
2^2
3^2
4^2
|
|
n

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

k th term $t_k = k^2$

$k^2 = n$

$k = n^{1/2}$

Time Complexity $= O(1+1+1+1+n^{1/2})$

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

$= O(\sqrt{n})$

Ques For the function n^k and c^n what is the asymptotic notation relationship btw these func. Assume $k>=1$ & $c>1$ are consts. find out the value of c and no for which relation holds.

As given n^k & c^n

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

as $n^k \le a \cdot c^n$ ∀ $n \geq n_0$ for a const $a>0$

for $n_0 = 1$

$c = 2$

$1^k \le a 2^1$

$\boxed{n_0 = 1 \ \& \ c = 2}$