Tutorial - 1

Ques 1) What do you understand by Asymptotic notation, define different asymptotic notation with example ?

i) Big O(n)

$f(n) \Rightarrow O(g(n))$

if $f(n) \leq g(n) \times c \;\forall\; n \geq n_0$

for some Constant, $c > 0$

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

eg:- $f(n) = n^2 + n$

$g(n) \Rightarrow n^3$

$n^2 + n \leq c * n^3$

$n^2 + n = O(n^3)$

(ii) Big Omega (Ω)

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

means $g(n)$ is "tight" lowerbound of $f(n)$ ie $f(n)$ can go beyond $g(n)$

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

if and only if

$f(n) \geq c . g(n) \quad \forall\; n_2 > n_0$ and c = Constant > 0

Ex:- $f(n) \Rightarrow n^3 + 4n^2$

$g(n) \Rightarrow n^2$

i.e $f(n) \geq c * g(n)$

$n^3 + 4n^2 = \Omega(n^2)$

(iii) Big Theta (Θ)

when $f(n) = \Theta(g(n))$, gives the tight upperbound & lowerbound both

i.e $f(n) = \Theta\, g(n)$

if $c_1 g(n_1) \leq f(n) \leq c_2 g(n_2)$

for all $n \geq \max(n_1, n_2)$ and some constant $C_1 > 0$ & $C_2 > 0$

Eg:- $3n+2 = \theta(n)$ as $3n+2 \geq 3n$

(iv) Small On(o)

when $f(n) = og(n)$ gives the upper bound.

i.e $f(n) = og(n)$

iff $f(n) < cg(n)$

$\forall n > n_0$ & $n > 0$

Ex:- $f(n) = n^2$; $g(n) = n^3$

$f(n) < g(n)$

$n^2 = o(n^3)$

V) Small Omega(ω):-

It gives the lower bound;

i.e $f(n) = \omega g(n)$

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

if $f(n) > cg(n)$ $\forall n > n_0$ and some const $c > 0$.

Ques 2) What should be the time complexity of

for(int i=1 to n)

{

$i = i*2$ — $O(1)$

}.

for $i = 1, 2, 4, 8, 16$ - - - - n times

So, $a=1$, $r = 2/1 = 2$ GP

kth value of GP;

$t_k = ar^{k-1}$

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

$2^n = 2^k$

$\log_2(2n) = k\log 2$

$\log_2 2 + \log_2 n = k$

$\log_2 n + 1 = k$

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

Que. 3) $T(n) = \begin{cases} 3T(n-1) & \text{if } n > 0, \\ \text{otherwise } 1 \end{cases}$

$T(n) = 3T(n-1)$ — (1)

$T(n) = 1$

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

$T(n-1) = 3T(n-2)$ — (2)

Put (2) in (1)

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

$T(n) = 9T(n-2)$ — (3)

Put $n = n-2$ in — (1)

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

Put in (3)

$T(n) = 27T(n-3)$ — (4)

$T(k) = 3^k T(n-k)$ — (5)

for kth term Let $n-k = 1$

$K = n-1$

Put in (5)

$T(n) = 3^{n-1}\, T(1)$

$= 3^{n-1}$

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

Ques-4) $T(n) = \begin{cases} 2T(n-1)-1 & \text{if } n>0, \\ \text{otherwise } 1 \end{cases}$

$T(n) = 2T(n-1)-1$ — (1)

Put $n = n-1$

$T(n-1) = 2T(n-2)-1$ — (2)

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

$= 4T(n-2)-2-1$ — (3)

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

Put in (1)

$T(n) = 8T(n-3)-4-2-1$ — (4)

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

K^{th} term

Let $n = K-1$

$K = n-1$

$T(n) = 2^{n-1} T(1) - 2^K\left(\frac{1}{2}+\frac{1}{2^2}+ \cdots \frac{1}{2^K}\right)$

$= 2^{n-1} - 2^{n-1}\left(\frac{1}{2}+\frac{1}{2^2}+ \cdots \frac{1}{2^{n-1}}\right)$ $a = \frac{1}{2}$, $r = \frac{1}{2}$

So

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

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

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

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

Ques-5) What should be time Complexity of

```
int i=1, S=1;
while(S<=n){
  i++; S=S+i;
  Print("#");
}.
```

$i = 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \; ---$

$S = 1+3+6+10+15+21+ \cdots$ — ①

Sum of $S = 1+3+6+10+ \cdots T_{n-1} + T_n$ — ②

$0 = 1+2+3+4+ --- n - T_n$

$T_K = 1+2+3+4+ ---- + K$

$T_K = \frac{1}{2}K(K+1)$

for K iterations

$1+2+3+ --- K \leq n$

$\frac{K(K+1)}{2} \leq n$

$\frac{K^2+K}{2} \leq n$

$O(K^2) \leq n$

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

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

Ques-6) Time Complexity of

```
Void f(int n)
{
  int i, count=0;
  for(int i=1; i<=n; i++)
}.
```

$i^2 = n$

$i = \sqrt{n}$

$i = 1, 2, 3, 4 \; --- \; \sqrt{n}$

$\sum_{i=1}^{n} = 1+2+3+4+ --- \sqrt{n}$

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

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

$T(n) = O(n)$

Ques 7) 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 ++
}.
```

Since for $K = K^2$

$K = 1, 2, 4, 8 \text{ --- } n$

$a = 1, \; r = 2$

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

$n = 2^k - 1$

$n + 1 = 2^k$

$\log_2(n) = K$

i	j	K
1	log n	log(n) * log(n)
2	log n	log(n) log(n)
3	log n	log(n) log(n)
⋮	⋮	⋮
n	log n	log(n) . log(n)

T.C = O(n * log n * log n)

$= \underline{O(n \log^2(n))}$ — Ans.

Ques 8) Time Complexity of

```
Void function(int n)
{
  if(n==1) return;
  for(i=1 to n)
   for(j=1 to n)
     Printf("*");
   }
  }
 function(n-3);
}.
```

Sol:- for(i=1 to n)

we get j n times every turn

$i * j = n^2$

Kth;

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

$T(n-3) = (n-3)^2 + T(n-6)$

$T(n-6) = (n-6)^2 + T(n-9)$

& $T(1) = 1$

Now Substitute each Value in T(n)

$T(n) = n^2 + (n-3)^2 + (n-6)^2 + \text{ --- } + 1$

Let

$K^n - 3K = 1$

$K = (n-1)/3$ total terms $= K+1$

$T(n) = n^2 + (n-3)^2 + (n-6)^2 + \cdots + 1$

$T(n) = Kn^2$

$T(n) = (K-1)/3 * n^2$

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

Ques-9) Time Complexity :-

for $i=1$ $j = 1+2+ \cdots n \geq j+1$

$i=2$ $j = 1+3+5+ \cdots n \geq j+1$

$i=3$ $j = 1+4+7+ \cdots n \geq j+1$

n^{th} term of AP is

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

$T(n) = 1 + (n-1)d$

$(n-1)/d = n$

for $i=1$ $(n-1)/1$ times

for $i=2$ $(n-1)/2$ times

!

$i = n-1$

we get

$T(n) = i_1 j_1 + i_2 j_2 + \cdots i_{n-1} j_{n-1}$

$= \frac{(n-1)}{2} + \frac{(n-2)}{2} + \frac{(n-3)}{3} + \cdots 1$

$= n + \frac{n}{2} + \frac{n}{3} + \cdots \frac{n}{n-1} - n*1$

$= n\left[1 + \frac{1}{2} + \frac{1}{3} + \cdots \frac{1}{n-1}\right] - n*1$

$= n\log n - n + 1$

Since

$\int \frac{1}{x} = \log x$

$\underline{T(n) = O(n\log n)}$ Ans

Qu-10) ____

Sol:- As given n^k & c^n

Relationship b/w n^k & c^n is

$n^k = O(c^n)$

$n^1 \leq a(c^n)$

$\forall\, n \geq n_0$ & constant, $a > 0$

for $n_0 = 1$; $c = 2$

$1^k < a2$

$\underline{n_0 = 1 \text{ \& } c = 2}$ An.