Samya Jain
40
I

Design and Analysis of Algorithm

Tutorial - 1

Que-1. What do you understand by asymptotic notations. Define different asymptotic notations with example.

Ans-1. Asymptotic notation means towards infinity. They are used to tell the complexity of an algorithm having input size very large.

It is priori analysis.

Different Types of Asymptotic notations are :-

① Big Oh Notation -

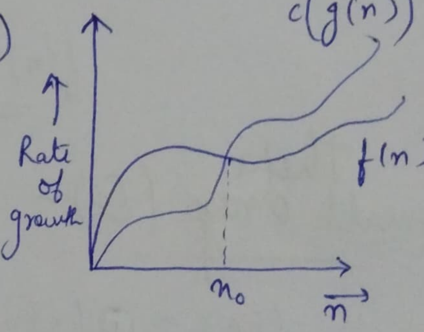
$f(n) = O(g(n))$, if $0 \leq f(n) \leq c(g(n))$ $\forall n \geq n_0$

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

Example -

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

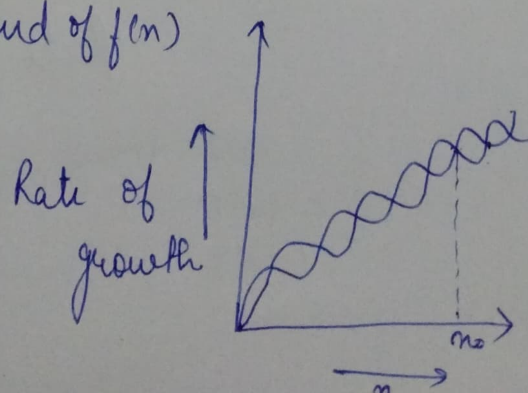
$T(n) = O(n)$



② 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 $f(n)$



(3) Big Omega (Ω) -

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

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

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

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

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

$c \leq 6 + \frac{1}{n} + \frac{1}{n^2}$

$C \leq 6$ { on putting $n = \infty$, $\frac{1}{n} = \frac{1}{\infty} = 0$

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

$6 \leq 6 + 1 + 1$

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

$n_0 = 1$

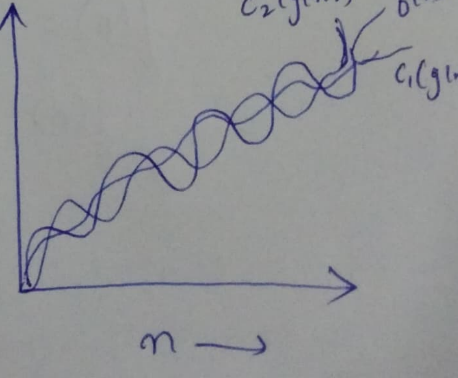
$f(n) = \Omega(n^2)$

(4) 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 & $c_2 \geq 0$

Rate of growth ↑

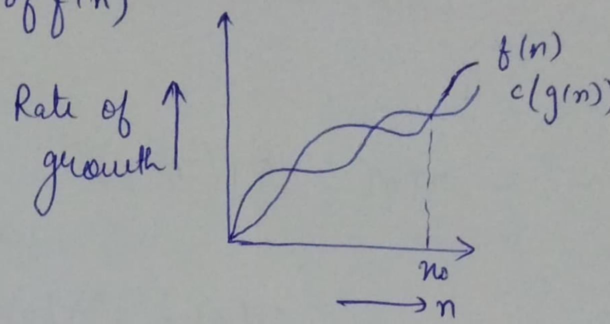


5

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 $f(n)$



6 **Que-2.** what should be time complexity of

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

Ans-2: I would have 1, 2, 4, 8, 16, ----, n.

Let say there 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 on both sides.

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

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

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

$$T(n) = O(k) = O(1 + \log n) \Rightarrow \underline{O(\log n)}$$ Ans.

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

Ans-3. $T(n) = 3T(n-1)$ — 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

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

⋮

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

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

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

$T(n) = 3^k \Rightarrow \boxed{T(n) = O(3^k)}$ Ans.

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

$T(n) = 2T(n-1) - 1$ — ①

By using back substitution method.

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

$= 2^2 T(n-2) - 2 - 3$ — ② $\qquad T(n-1) = 2T(n-2)-1$

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

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

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

⋮ Continues for k terms.

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

Assume $n-k=0 \Rightarrow n=k$ $\qquad T(0)=1$

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

$\Rightarrow 2^n - 2^{n-1} - 2^{n-2} \ldots -1$

$= 2^n - [2^{n-1} + 2^{n-2} \ldots +1]$

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

Sum of G.P. $= \dfrac{a(1-r^{n-1})}{1-r}$

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

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

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

$$\Rightarrow 2^{n}-[2^{n}-2] \Rightarrow 2$$

$$\therefore \Rightarrow T(n) = O(2)$$

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

Que-5. what should be the complexity of

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

i	s
1	1
2	3
3	6
4	10
5	15
⋮	⋮
n	n
	k times

$s = 1, 3, 6, 10, 15, ---- n$

Let say k terms

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

$$k^2 = 2n$$

$$\boxed{k = \sqrt{n}}$$

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

$\because t_{k-1}$ will be constant

$$\Rightarrow \boxed{T(n) = O(n)}$$ Ans.

Que-6. Time complexity of: void function (int n)

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

Ans-6 $1^2, 2^2, 3^2, \ldots n.$

let say k terms

$t_k = k^2$

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

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

Que-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 ++;
}
```

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

$= \frac{n}{2}, \frac{n+2}{2}, \frac{n+4}{2}, \ldots n.$

General form $\Rightarrow \frac{n+0*2}{2} + \frac{n+1*2}{2} + \frac{n+2*2}{2} + \ldots n$

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

Total terms = k+1

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

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

$n - 2 = 2k$

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

i	j	k
$\frac{n}{2}$	$\log n$ 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		

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

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

$\boxed{T(n) = O(n \log^2 n)}$ Ans.

Que-8. Time Complexity of :

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

Ans-8. function all would be $n, n-3, n-6, n-9, \ldots,$
let say, k terms.

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

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

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

$1 = n - 3k + 3$

$3k = n + 2$

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

$\because$ function have recursion calls $\frac{n+2}{2}$ times.

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

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

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

Que-9. Time complexity of:

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

Ans-9. for i (outer loop)

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

when $i = 2 \rightarrow j = 1, 3, 5, 7, \ldots n \Rightarrow n/2$

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

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

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

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

Que-10. for the function, n^k and c^n, what is the asymptotic relationship b/w these functions? Assume that k>=1 and c>1 are constants. find out the value of c and n0 for which relation holds.

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

relationship 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 $n_0 = 1$

$c = 2$

$1^k \leq a2$

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