Design and Analysis of Algorithms
Tutorial - 1

Name → Muskan Agarwal
Section → F
Roll no. → 15
University Roll no → 2016860

Q1- What do you understand by Asymptotic notations. Define different Asymptotic Notation with excamples.

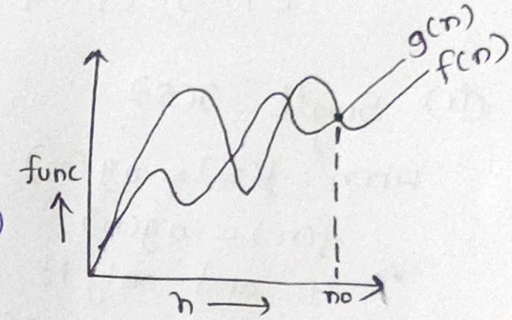
⇒ (i) Big O(n)

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

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

for some constant, $c > 0$

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



Ex → $f(n) = n^2 + n$

$g(n) = n^3$

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

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

(ii) Big Omega (Ω)

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

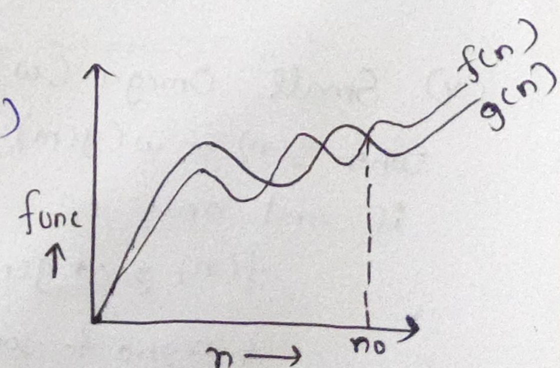
$g(n)$ is 'tight' lowerbound of $f(n)$

ie. $f(n)$ can go beyond $g(n)$

ie. $f(x) = \Omega\, g(x)$

if and only if

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



Ex → $f(n) \Rightarrow n^3 + 4n^2$

$g(n) = n^2$

ie. $f(n) \geq c * g(n)$

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

(iii) Big Theta (Θ)

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

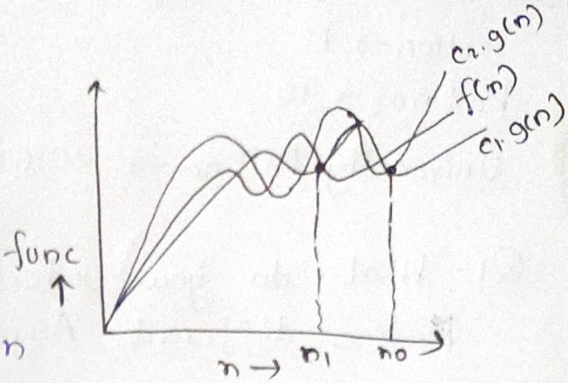
It gives tight upperbound and lowerbound both.

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

if and only if

$c_1 * g(n) \le f(n) \le c_2 * g(n_2)$

$\forall\ n \ge \max(n_1, n_2)$, some constant $c_1 \& c_2 > 0$.



Ex → $3n+2 = \Theta(n)$ as $3n+2 \ge 3n$

& $3n+2 \le 4n$ for n

$c_1 = 3$, $c_2 = 4$ & $n_0 = 2$

(iv) Small o(o)

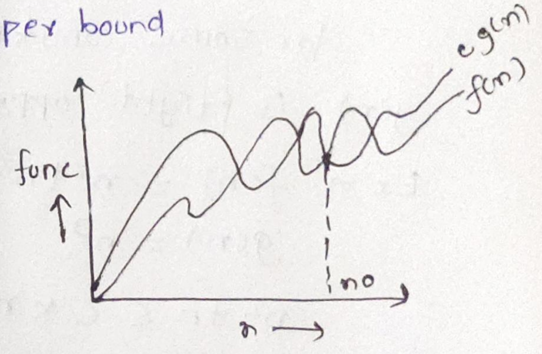
When $f(n) = og(n)$ gives the upper bound

$f(n) = og(n)$

if and only if

$f(n) < cg(n)$

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



Ex → $f(n) = n^2$; $g(n) = n^3$

$f(n) < c\, g(n)$

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

(v) Small Omega (ω)

When $f(n) = \omega(g(n))$ gives the lower bond

if and only if

$f(n) > c * g(n)$

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

For ex → $f(n) = n^2$ $g(n) = n$

$\therefore$ $n^2 = \omega(n)$

Q2- What should be time complexity of

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

⇒ $i \Rightarrow 1, 2, 4, 6, 8, 10 \ldots \ldots n$ times

It is an G.P.

$a=1 \quad r = \frac{2}{1} = 1$

To find the k^{th} term

$t_k = a \cdot r^{k-1}$

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

$n = \frac{2^k}{2}$

$2n = 2^k$

app. $\log_2 2$ on both sides

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

$\log_2 2 + \log_2 n = k \log_2 2$

$1 + \log_2 n = k$

$k = \log_2 n$

$\boxed{T.C. \Rightarrow O(\log_2 n)}$

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

$T(n) = 3T(n-1) \quad — ①$

$T(1) = 1$

put $n = n-1$ in ①

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

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

put the value in ①

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

$T(n) = 9T(n-2) \quad — ②$

put $n = n-2$ in 1

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

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

put the value in 2

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

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

Generalising Series

$T(n) = 3^k T(n-k)$ — 3

for kth terms, Let $n-k=1$

$k = n-1$

put in 3

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

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

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

Q4. $T(n) = \{2T(n-1) - 1$ if $n > 0$, otherwise $1\}$

⇒ $T(n) = 2T(n-1) - 1$ — 1

put $n = n-1$

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

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

put in 1

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

$T(n) = 4T(n-2) - 2 - 1$ — 2

Now put $n = n-2$ in 1

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

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

put in (2)

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

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

$$T(n) = 2^k T(n-k) - 2^{k-1} - 2^{k-2} - \dots \dots 2^0 \quad -(3)$$

k^{th} term → let $n-k=1$

$k = n-1$

Put $k=n-1$ in (3)

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

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

i.e. Series in G.P.

$a = 1/2 \qquad r = 1/2$

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

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

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

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

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

Q6 - int i = 1, s = 1;

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

⇒ $i = 1, 2, 3\ 4\ 5\ 6 \ldots$

$s = 1 + 3 + 6 + 10 + 15 + \ldots + T_{n-1} + T_n$

$T_k = 1+2+3+4+ \ldots + k$

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

For k iterations

$1+2+3 \ldots k <= n$

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

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

$O(k^2) <= n$

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

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

Q6 - Time complexity of -

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

ol -

$i^2 = n$

$i = n$

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

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

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

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

Q7 - 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++;
}
```

⇒ For j & k

1, 2, 4, 8, 16 .... n

$2^0, 2^1, 2^2, 2^3, 2^4 .... 2^k$

It is in G.P.

$a=1 \quad r=2$

$t_k = \cancel{a r^{k}} \; a \cdot r^{k-1}$

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

$n = \frac{2^k}{2}$

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

$\log_2 2 + \log_2 n = k \log_2 2$

$k = \log_2(n)$

i	j	k
1	log(n)	log(n)
2	log(n)	log(n)
⋮	⋮	
n	log(n)	log(n)

$T.C. = n \times \log(n) \times \log(n) \Rightarrow \boxed{T(n) = O(n(\log_2^2 n))}$

Q8 - Time complexity of

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

⇒ for (i=1 to n)

we get $j=n$ times every time

$\therefore i*j = n^2$

Now, $T(n) = n^2 + T(n-3);$

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

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

and $T(1) = 1$

Substitute each value in $T(n)$

$T(n) = n^2 + (n-3)^2 + (n-6)^2 + T(n-9)$

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

Let $n - 3k = 1$

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

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

$T(n) \simeq kn^2$

$T(n) \simeq (k-1)/3 \quad n^2$

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

Q9 - Time Complexity of

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

$\Rightarrow$ For $i = 1$ $\quad j = 1 + 2 \ldots\ldots$ $(n \geq j + i)$

$i = 2$ $\quad j = 1 + 3 + 5 \ldots$

$i = 3$ $\quad j = 1 + 4 + 7 \ldots$

n^{th} term of AP is

$T(n) = a + d \times n$

$T(m) = 1 + d \times n$

$(n-1)/d = n$

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

$i = 2$ $\quad (n-1)/2$ times

$i = n-1$

we get,

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

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

$= n + \frac{n}{2} + \frac{n}{3} + \ldots + \frac{n}{n-1} \quad - n \times 1$

$= n\left[1 + 1/2 + \frac{1}{3} + \ldots + \frac{1}{n-1}\right] - n \times 1$

$= n \times \log n - n + 1$

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

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

Q10- For the functions, n^k and c^n, what is the asymptotic relationship between these functions? Assume that $k >= 1$ and $c > 1$ are constants. Find out the value of c and no for which relation holds.

→ As given n^k and c^n

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

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

$$n^k \leq a\,(c^n)$$

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

for $n_0 = 1$, $c = 2$

$$1^k < a^2$$

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