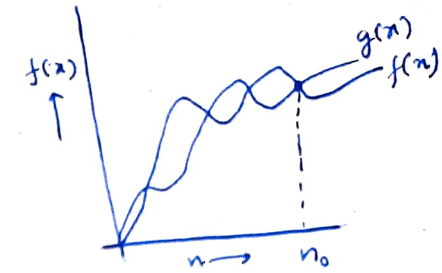
Q.1 What do you understand by Asymptotic Notation, define different asymptotic notation with example. 1

TUTORIAL-1

(i) **Big Oh (O)**



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

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

for some constant, $c > 0$

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

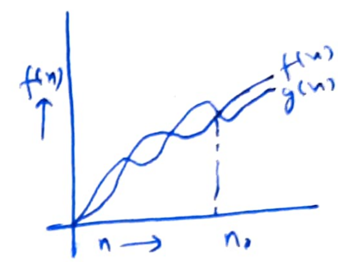
Eg:- $f(n) = n^2+n$

$g(n) = n^3$

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

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

(ii) **Big Omega (Ω)**



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

means $g(n)$ is 'tight' lower bound of $f(n)$; i.e., $f(n)$ can go beyond $g(n)$

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

iff $f(n) \geq c\,g(n) \quad \forall\ n > n_0$ and some constant $c > 0$

Eg:- $f(n) = n^3 + 4n^2$

$g(n) = n^2$

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

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

(iii) **Big Theta (θ)**

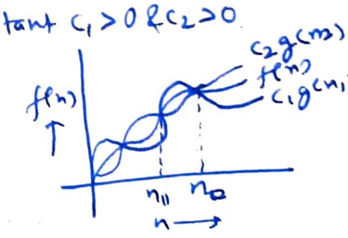
When $f(n) = \theta\, g(n)$, gives the tight upperbound and lowerbound both.

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

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

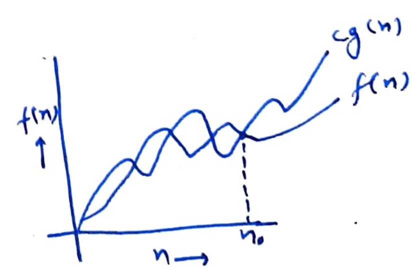
for all $n \geq \max(n_1, n_2)$ and some constant $c_1 > 0$ & $c_2 > 0$

i.e., $f(n)$ can never go beyond $c_2\, g(n)$ and will never come down of $c_1\, g(n)$.



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

(iv) **Small Oh (o)**



when $f(n) = o\,g(n)$ gives the upper bound

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

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

$\forall\ n > n_0$ and $n > 0$

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

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

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

Akashkm.

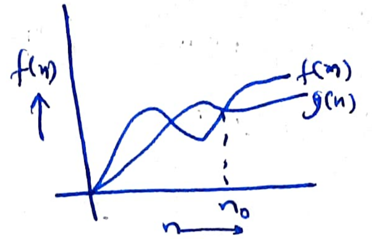
(v) Small Omega (ω) :-

It gives the lower bound ;

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

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

iff $f(n) > c g(n)$ $\forall$ $n > n_0$ and some const. $c > 0$



Q.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

G.P.

so, $a = 1$, $r = 2/1 = 2$

k^{th} value of GP:

$t_k = ar^{k-1}$

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

$2n = 2^k$

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

$\log_2 2 + \log_2 n = k$

$\log_2 n + 1 = k$

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

Q.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

Akanksha.

$T(K) = 3^k T(n-k)$ — ⑤

for k^{th} term, Let $n-k=1$

$k = n-1$

Put in ⑤

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

$= 3^{n-1}$

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

Q.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

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

k^{th} term:-

Let $n = k-1$

$k = n-1$

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

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

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

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

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

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

$T(n) = O(1)$

Akanksha.

Q.5) What should be time complexity of

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

$i = 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad \ldots$ ——— ①

$s = 1 + 3 + 6 + 10 + 15 + 21 + \ldots$

sum of s = $1+3+6+10+\ldots \quad T_{n-1} + T_n$ ——— ②

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

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

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

for k iterations

$1+2+3+\ldots k \le n$

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

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

$O(k^2) \le n$

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

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

Q.6) Time Complexity of

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

$i^2 = n$

$i = \sqrt{n}$

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

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

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

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

$T(n) = O(n)$

Akash

Q.7) Time complexity of

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

Since for $k = k^2$

$k = 1, 2, 4, 8, \ldots\ldots n$

$a = 1, \ r = 2$

$$\frac{a(r^n - 1)}{r-1}$$

$$\frac{1(2^k - 1)}{1}$$

$n = 2^k - 1$

$n + 1 = 2^k$

$\log_2(n) = k$

i	j	k
		$\log(n) * \log(n)$
1	$\log n$	$\log(n)\log(n)$
2	$\log n$	$\log(n)\log(n)$
3	$\log n$	$\log(n)\log(n)$
⋮	⋮	⋮
n	$\log n$	$\log(n)\log(n)$

$TC \Rightarrow O(n \log(n) \log(n))$

Q.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-1);
}
```

Akarsh

for (i=1 to n)
we get j n times every turn
$i * j = n^2$

k^{th};

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

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

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

and $T(1) = 1$

Now, substitute each value in $T(n)$

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

let $n - 3k = 1$

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

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

$T(n) \cong k\, n^2$

$T(n) = (k-1)/3 \; n^2$

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

Q.9) Time complexity of
void function (int n)
{
for (int i = 1 to n)
for(int j = 1 ; j <= n ; j += i)
printf("*");

}

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

Akanksha

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

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

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

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

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

Since

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

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

Q.10) For the function n^k & c^n. What is the asymptotic relationship b/w these functions?

Assume that $k >= 1$ & $c > 1$ are constants. Find out the value of c and no. of which relationship holds.

As given n^k & c^n.

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

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

$$n^k \le a(c^n)$$

$$\forall \quad n \ge n_0 \quad \& \text{ constant}, \; a > 0$$

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

$$1^k < a^2$$

$$n_0 = 1 \quad \& \quad c = 2$$

Akanksha.