Tutorial - 1

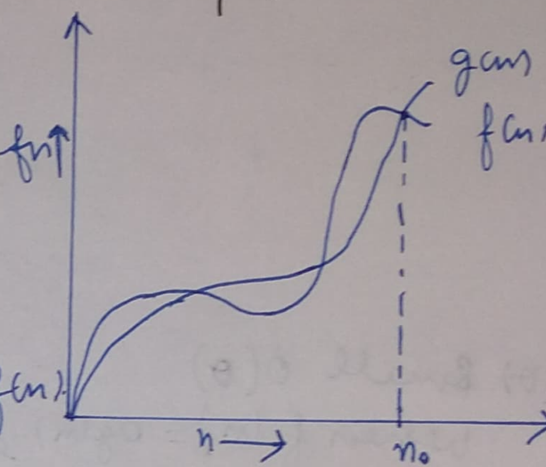
Name: HARSH RASTOGI   Class Roll No - 6
Section: F   University Roll No - 2016761

1. What do you underetand by Asymptotic Notations, define different with example.

Ans:- (i) Big $O(n)$

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

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 $f(n)$

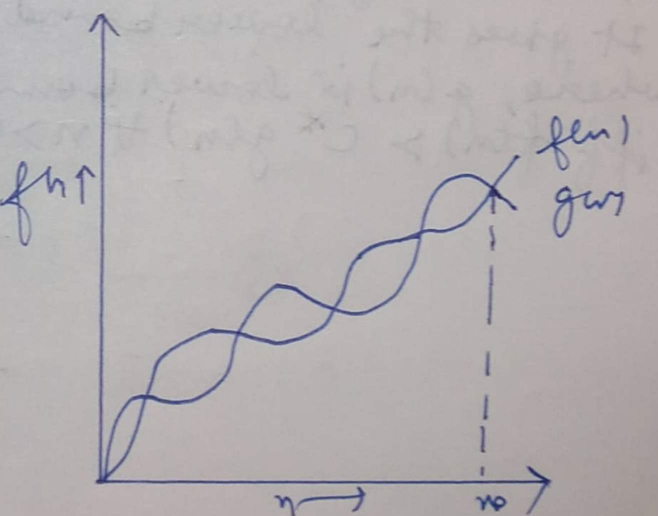


Eg → $f(n) = n^2+n$
$g(n) = n^3$

$n^2+n \leq C * n^3$
$n^2+n = O(n^3)$

(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 be going beyond $g(n)$ i.e.
$f(n) = \Omega\, g(n)$
iff $f(n) \geq g(n)$
$\forall\, n_2 > n_0$ & C is constant.

Eg ~ $f(n) = n^3 + 4n^2$
$g(n) = n^2$
i.e. $f(n) \geq C * g(n)$
$n^3 + 4n^2 \geq C * n^2$
$n^3 + 4n^2 = \Omega(n^2)$

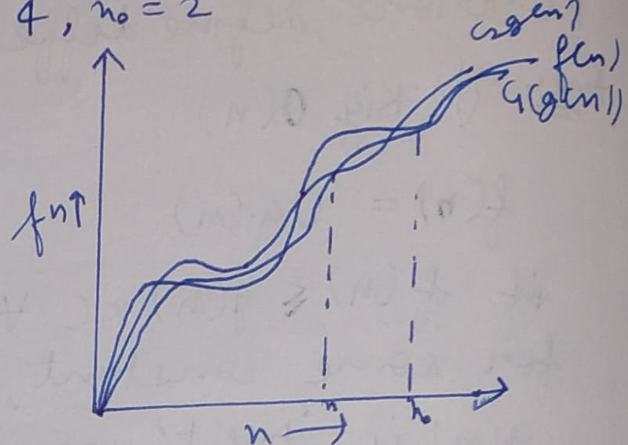


(iii) Big Theta Θ

When $f(n) = \Theta(g(n))$ gives the tight upper bound & lower bound both. i.e. $f(n) = \Theta(g(n))$ iff $C_1 * g(n_1) \le f(n) \le C_2 * g(n_2)$ for all $n \ge \max(n_1, n_2)$, some constant $C_1 > 0$ & $C_2 > 0$, i.e. $f(n)$ can never go beyond $C_2 g(n)$ & will never come down of $C_1 g(n)$.

Eg- $3n+2 = \Theta(n)$ as $3n+2 \ge 3n$ &

$3n+2 \le 4n$ for n, $C_1 = 3, C_2 = 4, n_0 = 2$



(iv) Small $o(o)$

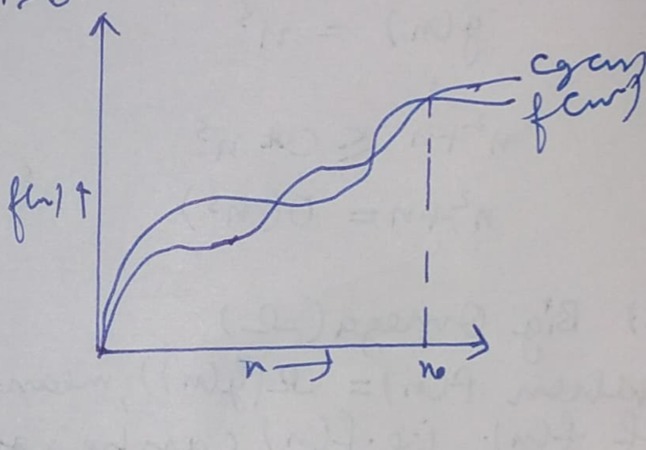
When $f(n) = o g(n)$ gives the upper bound if $f(n) = o g(n)$.

iff $f(n) < C * g(n)$ $\forall n > n_0$ & $n > 0$.

Eg→ $f(n) = n^2$; $g(n) = n^3$

$f(n) < C * g(n)$

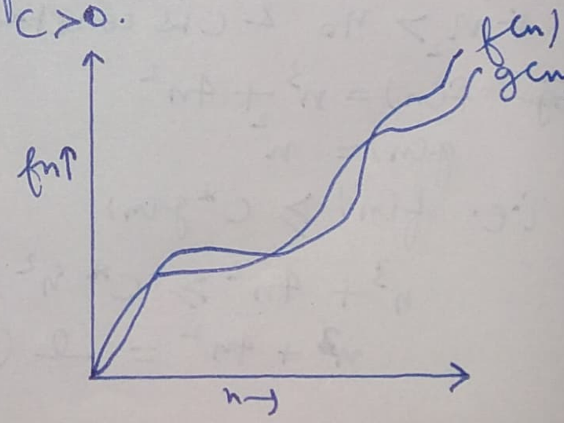
$n^2 = O(n^3)$



(v) Small Omega (ω)

It gives the 'lower bound' i.e. $f(n) = \omega(g(n))$ where, $g(n)$ is lower bound of $f(n)$

iff $f(n) > C * g(n)$ $\forall n > n_0$ & $C > 0$.



2. What should be the time complexity of

for (int i=1 to n)

{

i = i * 2; ⟶ O(1)

}

Ans:- for $i = 1, 2, 4, \ldots\ldots$ n times

i.e. series is a G. P.

So, $a = 1, r = 2$

K^{th} value of G.P.

$t_K = a r^{K-1}$

$t_K = 1(2)^{K-1}$

$2n = \cancel{2*} 2^K$

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

$\log_2 2 + \log_2 n = K$

$\log_2 n + 1 = K$

So, time complexity is $T(n) \Rightarrow O(\log_2 n)$

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

Ans:- $T(n) \Rightarrow 3T(n-1)$ —— (1)

$T(n) = 1$

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

$T(n-1) = 3T(n-1-1)$ —— (2)

put (2) in (1)

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

$T(n) = 9T(n-2)$ — 3

Put $n = n-2$ in 1

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

put it in 3

$T(n) = 27T(n-3)$ — 4

So, $T(k) = 3^k T(n-k)$ — 5

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

$k = n-1$, put in 5

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

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

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

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

Ans: $T(n) = 2T(n-1) - 1$ — 1

put $n = n-1$

$T(n-1) = 2T(n-2) - 1$ — 2

put in 1

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

$= 4T(n-2) - 2 - 1$ — 3

put $n = n-2$ in 1

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

put in 1

$T(n) = 8T(n-3) - 4 - 2 - 1$ — 4

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

$\Rightarrow$ k^{th} Term,

let $n-k = 1$

$k = n-1$

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

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

$\therefore$ a. G.P.

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

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

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

$T(n) = O(1)$

5.
```
int i=1, s=1;
  while (s<=n)
  {
      i++;
      s = s+i;
      printf("#");

  }
```

Ans:- $i = 1, 2, 3, 4, 5 \ldots\ldots$

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

Sum of $s = 1 + 3 + 6 + 10 + \ldots\ldots n$ — (1)

Also, $s = 1 + 3 + 6 + 10 + \ldots\ldots T_{n-1} + T_n$ — (2)

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

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

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

for k, $1+2+3+ \cdots\cdots k \leq n$

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

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

$$O(k^2) \leq n$$

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

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

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

Ans:- As $i^2 = n$

$i = \sqrt{n}$

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

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

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

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

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

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

Ans:- Since for $n = K^2$

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

$\because$ series is in G.P.

So, $a = 1, r = 2$.

$$a \cancel{\times} \frac{(r^2 - 1)}{r - 1}$$

~~a x 1~~

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

$n = 2^K - 1$

$n + 1 = 2^K$

$\log_2(n) = K$

i	s	K
1	log(n)	log(n) * log(n)
2	log(n)	log(n) * log(n)
⋮	⋮	⋮
n	log(n)	log(n) * log(n)

$T \cdot C \Rightarrow O(n * \log n * \log n)$
$\Rightarrow O(n \log^2(n))$

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

Ans:- for (i=1 to n)
we get j=n times every time
$i * j = n^3$

K^{th}, now,
$T(n) = n^2 + T(n-3);$
$T(n-3) = (n^2 - 3)^2 + T(n-6);$
$T(n-6) = (n^2 - 6)^2 + T(n-9);$
and $T(1) = 1;$

Now these values in $T(n)$
$T(n) = n^2 + (n-3)^2 + (n-6)^2 + \dots \dots + 1$

let $n - 3k = 1$
$k = (n-1)/3$
Total terms, $k+1$

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

$T(n) = k n^2$

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

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

9. void f(int n) {
for (int i = 1 to n) {
for (int j = 1; j <= n; j = j+i) {
printf (" * ");
}
}
}

Ans: for $i=1$ $\quad j = 1+2+3+\dots\dots\dots (n \geq j+i)$

$i=2$ $\quad j = 1+3+5+\dots\dots\dots (n \geq j+i)$

$i=3$ $\quad j = 1+4+7+\dots\dots\dots (n \geq j+i)$

n^{th} term of A.P is.

$T(n) = a + d*m$

$T(n) = 1 + d*m$

$(n-1)/d = m$

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

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

$i=n-1$

we get,

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

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

$= n + \frac{n}{2} + \frac{n}{3} + \dots\dots\dots n/n-1 + n \times 1$

$= n [1 + 1/2 + 1/3 + \ldots\ldots\ldots 1/n-1] - n*1$

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

Since, $\int 1/n = \log n$

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

---

10. For the function n^k & c^n, what is the asymptotic relationship b/we these functions. Assume that $k >= 1$ & $c > 1$ are constants. Find out the value of c & n_0 of which relationship holds.

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

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

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

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

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

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

$\Rightarrow 1^k = a^2$

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