Tutorial - 1

Name - Divyanshu
Section - F
Roll NO - 52
Univ Roll NO - 2016740

Q1 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' upperbound of f(n)

eg $f(n) \Rightarrow n^2 + n$

$g(n) \Rightarrow n^3$

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

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

ii) Big Omega(Ω)

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

means g(n) is "tight" lowerbound of f(n) i.e f(n) can go beyond g(n)

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

$\forall \ n \geq n_0$ and $C = \text{constant} > 0$

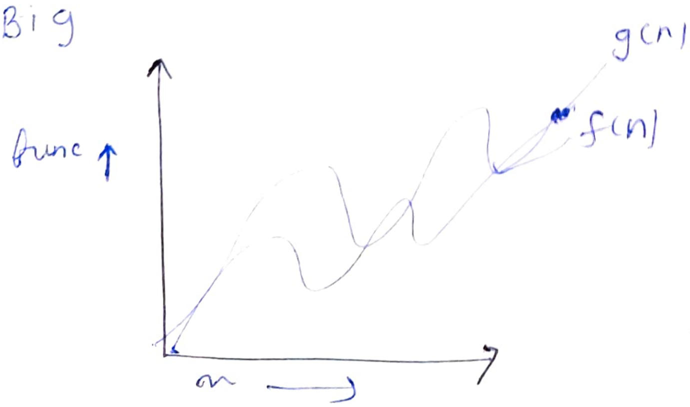
Ex $f(n) \Rightarrow n^3 + 4n^2$

$g(n) = n^2$

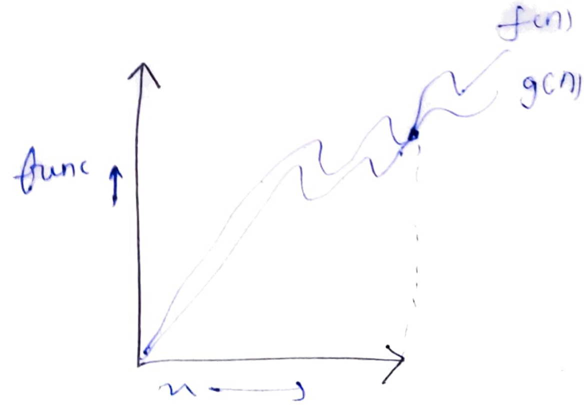
i.e $f(n) \geq C * g(n)$

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

iii) Big



Big(O)

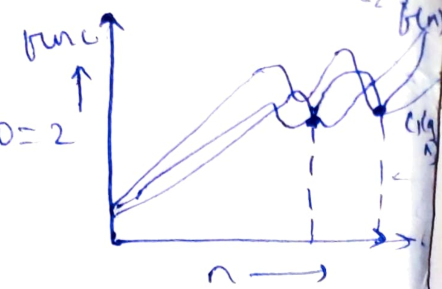


Big (Ω)

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

iv) $f(n)$ can go beyond $c_2(gn)$ and will never come down of $c_1 g(n)$

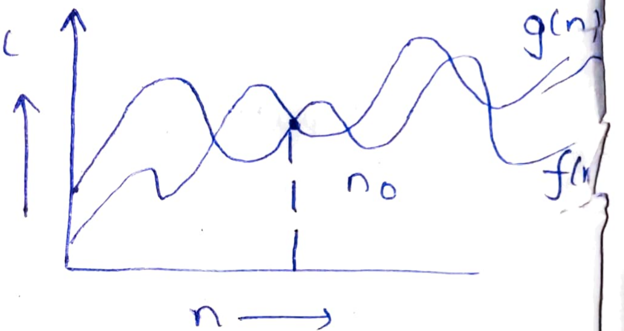
Ex $3n+2 = \theta(n)$ as $3n+2 \geq 3n$ El
$3n+2 \leq 4n$ for n, $c_1 = 3$ $c_2 = 4$ & $n_0 = 2$



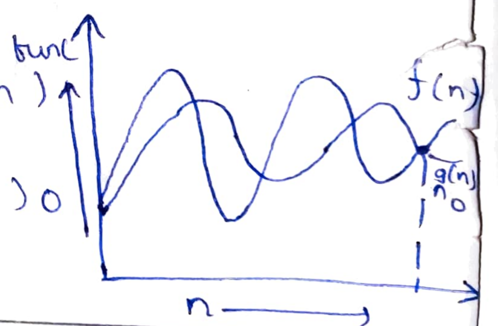
iv) small $o(\theta)$

when $f(n) = o\,g(n)$ given the upperhand
i.e. $f(n) = o\,g(n)$
if and only if
$f(n) > n_0$ & $n > 0$

Ex $f(n) = n^2$; $g(n) = n^3$
$f(n) < c * g(n)$
$n^2 = o(n^3)$



v) Small omega (n)
It gives the 'lower bound' i.e.
$f(n) = \omega(g(n))$
where $g(n)$ is lower bound of $f(n)$
if and only if $f(n) > c * g(n)$
$\forall\, n > n_0$ & some constant, $c > 0$



Q2.) what should be the time complexity of

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

↳ for $i \Rightarrow 1, 2, 4, 6, 8 \ldots n$ times
i.e. series is a G.P.
so $a = 1$, $r = 2/1$

k^{th} value of G.P.

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

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

$2^n = 2^k$

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

$\log 2 + \log n = K$

$\log_2 n + 1 = K$ (Neglecting '1')

So, Time complexity $T(n) \Rightarrow O(\log n)$ → Ans

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

otherwise 1

}

↳ i.e. $T(n) \Rightarrow 3T(n-1)$ —(1)

$T(n) \Rightarrow 1$

put $n \Rightarrow n-1$ in (1)

$T(n-1) \Rightarrow 3T(n-2)$ —(2)

putting (2) in (1)

$T(n) \Rightarrow 3 \times 3T(n-2)$ —(3)

putting $n \Rightarrow n-2$

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

put in (3)

$T(n) = 27T(n-3)$ —(4)

generalizing series

$T(K) = 3^K T(n-K)$ —(5)

for kth terms, let $n-K=1$ < Base

$K = n-1$

put in (5)

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

$T(n) = 3^{n-1}$ (neglecting 3')

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

Q4) $T(n) = [2T(n-1) - 1$ if $n>0$,

otherwise 1

]

$T(n) = 2T(n-1) - 1$ —(1)

put $n = n-1$

$T(n-1) = 2T(n-2) - 1$ —(2)

put in (1)

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

$= 4T(n-2) - 2 - 1$ —(3)

put $n = n-2$ in (1)

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

put in (1)

$T(n) = 8T(n-3) - 4 - 2 - 1$ — (4)

Generalising series

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

$\Rightarrow$ <u>k^{th} term</u> Let $n-k = 1$

$k = n-1$

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

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

ie series in G.P

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

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

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

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

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

Q: what should be Time complexity of

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

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

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

sum of $S = 1 + 3 + 6 + 10 + \cdots + T_n$ — (1)

Also $S = 1 + 3 + 6 + 10 + \cdots T_{n-1} + T_n$ — (2)

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

$T_K = 1 + 2 + 3 + 4 + \cdots + K$

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

for K iteration

$1 + 2 + 3 + \cdots K <= n$

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

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

$O(K^2) <= n$

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

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

Q6 Time complexity of

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

↳ As $i^2 = n$

$i = \sqrt{n}$

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

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

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

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

$T(n) = O(n)$ — Ans

Q7 Time complexity of

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

↳ since, for $k = n^2$

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

series is in G.P

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

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

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

$n = 2^k - 1$

$n+1 = 2^k$

$\log_2(n) = k$

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

T.C $\Rightarrow O(n * \log n * \log n)$

$\Rightarrow O(n \log^2(n))$ — Ans

Q8 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);
}
```

↳ for ($i=1$ to n)

we get $j = n$ times every term

∴ $i * j = n^2$

Now,

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

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

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

and $T(1) = 1$

Now substitute each value in $T(n)$

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

Let

$k^n - 3k = 1$

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

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

$T(n) \simeq kn^2$

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

So

$T(n) = O(n^3)$ → Ans

Q9 Time complexity of

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

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

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

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

n^{th} term of AP is

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

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

$(n-1)/d = n$

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

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

$i = n-1$

we get,

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

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

$= n + n/2 + n/3 + \cdots n/_{n-1} \cdots n \times 1$

$= n[1 + 1/2 + 1/3 + \cdots 1/_{n-1}] - n + 1$

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

Since $f(1/n) = \log x$

$T(n) = O(n \log n)$ — Ans

For the function $n^{-1}k$ & c^n. what is the asymtotic Relationship b/w - three function?

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

↳ As given n^k and c^n

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

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

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

$\forall n \geq no$ & constant, $a > 0$

for $no = 1, c = 2$

$\Rightarrow 1^k < a^2$

$\Rightarrow no = 1$ & $c = 2$ - Ans