Date / /

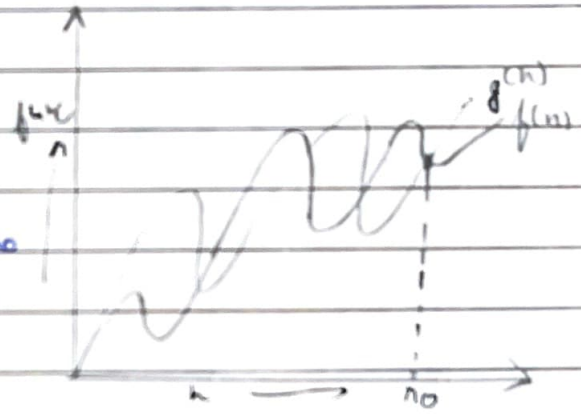
Tutorial - 1

Name - Anjali kohli
Section - F
Roll no. - 53
Univ. Roll no. - 2017718

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

(i) Big O (n)

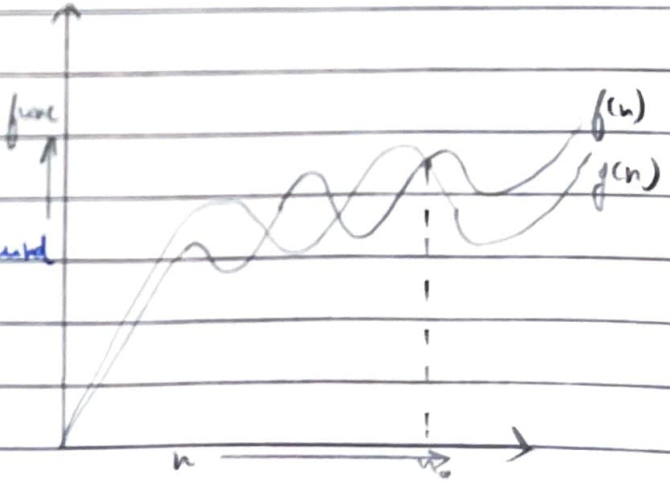
$f(n) \Rightarrow O(g(n))$
if $f(n) \leq g(n) * C \ \forall \ n \geq n_0$
for some constant, $C > 0$
$g(n)$ is 'tight' upper bound of $f(n)$



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" lower-bound of $f(n)$ ie $f(n)$ can go beyond $g(n)$
ie $f(x) = \Omega g(x)$



if

Date / /

if and only if
$f(n) \geq c \cdot g(n)$
$\forall \; n_2 > n_0$ and c = constant > 0

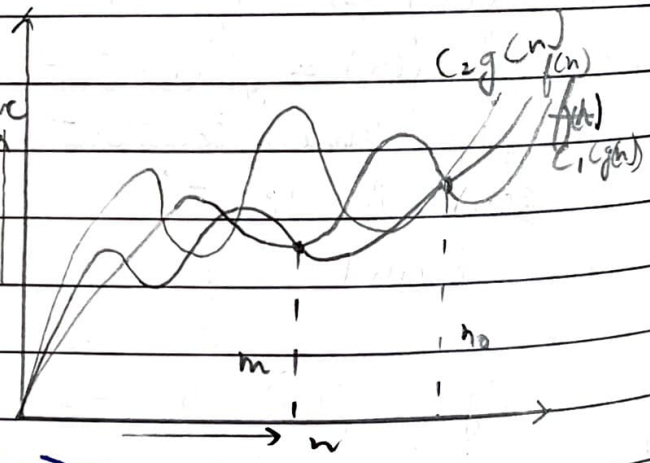
Ex - $f(n) \Rightarrow n^3 + 4n^2$
$g(n) \Rightarrow n^2$

ie $f(n) \geq c * g(n)$
$n^3 + 4n^2 = \Omega(n^2)$

(iii) Big Theta (Θ)

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

ie $f(n) = \Theta(g(n))$
if and only if
$c_1 * g(n_1) \leq f(n) \leq c_2 * g(n_2)$
for all $n \geq \max(n_1, n_2)$,
some constant $c_1 > 0$ & $c_2 > 0$



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

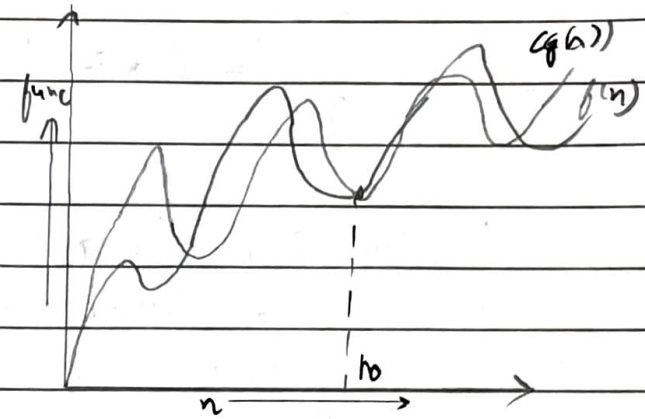
E:- $3n + 2 = \Theta(n)$ as $3n + 2 \geq 3n$ and
$3n + 2 \leq 4n$ for n, $c_1 = 3$, $c_2 = 4$ and $n_0 = 2$.

Date / /

(iv) Small o(o)

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

if and only if
$f(n) < c * g(n)$
$\forall$ $n > n_0$ and $n > 0$

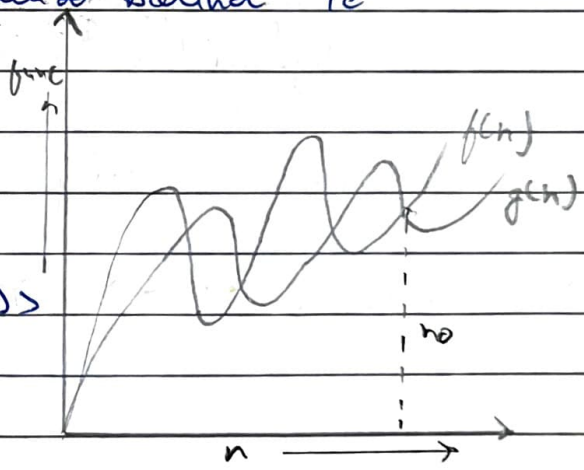


Ex:- $f(n) = n^2$; $g(n) = n^3$
$f(n) < c * g(n)$
$n^2 = o(n^3)$

(v) Small Omega (ω)

It gives the 'lower bound' ie

$f(n) = \omega(g(n))$
where $g(n)$ is lower bound of $f(n)$
if and only if $f(n) > c * g(n)$
$\forall$ $n > n_0$ and some constant, $c > 0$



Date / /

Q-2 What should be time complexity of :

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

for $i \Rightarrow 1, 2, 4, 6, 8 \ldots \ldots$ n times
ie Series is a GP
So $a = 1$, $r = 2/1$
k^{th} value of GP :
$t_k = a\, r^{k-1}$
$t_n = 1(2)^{k-1}$
$2n = 2^k$
$\log_2(2n) = k \log 2$
$\log_2 2 + \log_2 n = k$
$\log_2 n + 1 = k$ (Neglecting '1')

So, Time Complexity $T(n) \Rightarrow O(\log_2 n)$ → Ans

Q-3 $T(n) = \begin{cases} 3T(n-1) & \text{if } n > 0 \\ \text{otherwise } 1 \end{cases}$

ie $T(n) \Rightarrow 3T(n-1)$ — (1)
$T(n) = 1$
put $n \Rightarrow n-1$ in — (1)

$T(n-1) \Rightarrow 3T(n-2)$ — (2)
put (2) in (1)
$T(n) \Rightarrow 3 \times 3T(n-2)$
$T(n) \Rightarrow 9T(n-2)$ → (3)

Date / /

put $n \Rightarrow n-2$ in (1)

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

put in (3)

$T(n) = 2 \cdot 3T(n-3) \rightarrow (4)$

Generating series,

$T(k) = 3^k T(n-k) \quad - (5)$

for k^{th} terms, let $n-k=1$ (Base Case)

$k = n-1$

put in (5)

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

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

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

Q4 $T(n) = \begin{cases} 2T(n-1)-1 & \text{if } n>0, \\ 1 & \text{otherwise} \end{cases}$

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

put $n = n-1$

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

put in (1)

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

$= 4T(n-2) - 2 - 1 \quad - (3)$

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

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

Put in (1)

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

Date / /

Generalizing series

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

* k^{th} Term

$= 2^{n-1} T(1) - 2^k \left(\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} \dots \frac{1}{2^n}\right)$

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

ie series in G.P.

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

So,

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

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

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

$T(n) = O(1)$ Ans

Date / /

Q5 What should be the complexity of

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

i = 1 2 3 4 5 6 ...

s = 1 + 3 + 6 + 10 + 15 + ...

Sum of s = 1 + 3 + 6 + 10 + ... + n — (1)

Also s = 1 + 3 + 6 + 10 + ... T_{n-1} + T_n — (2)

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

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

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

for K iterations

$1 + 2 + 3 + \dots + K \leq n$

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

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

$O(K^2) \leq n$

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

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

Date / /

Q-6 Time Complexity of

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

As $i^2 = n$

$i = \sqrt{n}$

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

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

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

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

$T(n) = O(n)$ → Ans

Q-7 Time Complexity of

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

Date / /

Since, for $k = k^2$

$k = 1, 2, 4, 8 - - - - n$

∴ series is in GP

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 => $O(n * \log n * \log n)$

=> $O(n \log^2(n))$ Ans

Date /

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-3);
}
```

for (i=1 to n)

we get j=n times every turn

∴ i*j = n²

kth,

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

$k^2 - 3k = 1$

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

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

$T(n) \cong K n^2$

$T(n) \cong (K+1)/3 + n^2$

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

Date / /

Q-9 Time complexity of :-

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

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

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

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

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,

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

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

$= n + n/2 + n/3 + \dots n/(n-1) \dots 4 \times 1$

$= n [1 + 1/2 + 1/3 + \dots 1/n - 1] \dots n \text{ terms}$

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

Since $\int 1/x = \log x$ $T(n) = O(n \log n)$ Ans

Date / /

Q-10 For the function n^k & c^n, what is the asymptotic Relationship b/w these functions?

Assume that $k >= 1$ & c are constants. Find the value of c and n_0 of which relationship holds.

As given n^k and c^n

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

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

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

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

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

$$\Rightarrow 1^k \leq a^2$$

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