Name - Anshak Gupta
Sec - I
University roll no :- 2016648

TUTORIAL - 1

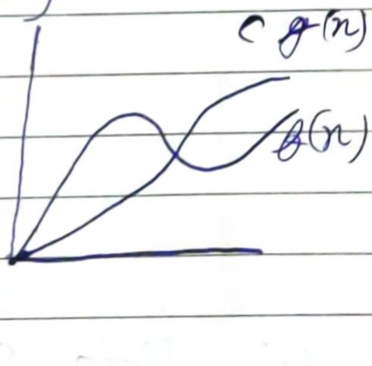
Q1 What do you understand by asymptotic notation, define different asymptotic notation with examples.

Ans1 Asymptotic notation means towards infinity. They are used to tell the complexity of an algorithm.
Having input size very large.
It is priority analysis.
Different types of asymptotic notations are:

① Big Oh notations:

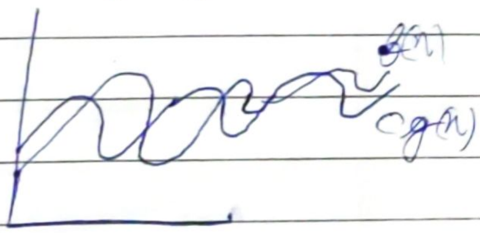
$f(n) = O(g(n))$
if $0 \le f(n) \le c(g(n)) \ \forall \ n > n_0$
$g(n)$ is tight upper bound of $f(n)$.

Example:
```
for(int i=0; i<n; i++)
{
  Cout << i << endl;
}
```
$T(n) = O(n)$



② Small Oh notation:

$f(n) = o(g(n))$, if $f(n) < c(g(n))$
$\forall \ n > n_0 \ \forall c > 0$
$g(n)$ is upper bound of $f(n)$



classmate
Date
Page

③ Big Omega (Ω)

$f(n) = \Omega\ g(n)$ if $f(n) = C(g(n)) \cdot \forall n \geq n_0$

Some constant $c > 0$

$g(n)$ is tight lower bound of $g(n)$.

Example $f(n) = 6n^2 + n + 1$

$0 \leq g(n) \leq f(n)$

$0 \leq c \cdot g(n) \leq f(n)$

$c \leq 6 + \frac{1}{n} + \frac{1}{n^2}$ on putting $n = \infty, \frac{1}{n\infty} = 1$

$c \leq 6 \quad 6 \leq 6 + 1 + < 6 + 0$ True

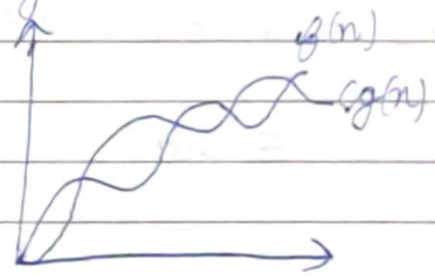
$c > 0$ and $n > n_0$ $(n = 1)$

$f(n) = \Omega(n^2)$

④ Small Omega (ω)

$f(n) = \omega(g(n))$, if $f(n) > c(g(n))\ \forall n > n_0$

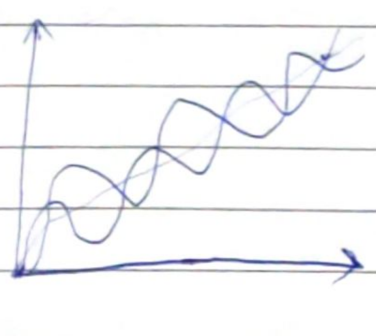
$g(n)$ is the lower bound of $f(n)$



⑤ Theta (Θ)

$f(n) = \Theta(g(n))$, if $c_1(g(n)) \leq f(n) \leq c_2(g(n))$

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



Q2 What should be time complexity of
for (i=1 to n) (i=i*2)

Ans 2 i would have 1, 2, 4, 8, 16, . . . n
let say there are k terms
It is a G.P with $a=1, r=2$
k^{th} term $= t_k = ar^{k-1}$
$n = 1(2)^{k-1}$
$= 2^{k-1}$

taking $\log_2$ on b/s
$\log_2 n = \log_2 (2^{k-1})$
$\log_2 n = (k-1)\log_2 2$
$\log n = (k-1) = k = 1+\log_2 n$
$T(n) = O(k) = O(1+\log n) = O(\log n)$

Q3 $T(n) = [3T(n-1)$ if $n>0$, otherwise

$T(n) = 3T(n-1)$ — ①
by backward substitution
$T(n) = 3T(n-1)$
$T(n-1) = 3T(n-1-1)$
$T(n-1) = 3T(n-2)$ — ②
Put ② in ①
$T(n) = 3[3T(n-2)] \Rightarrow T(n) = 9T(n-2)$ — ③
$T(n-2) = 3T(n-3)$
$T(n) = 2T(n-3)$
Continue for k times
$T(n) = 3^k T(n-k)$
assume $n-k=0 \Rightarrow n=k$
$T(n) = 3^k T(0)$
$T(n) = 3^k$
$T(n) = O(3^n)$

classmate

Date

Page

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

Ans 4 $T(n) = 2T(n-1) - 1$ — ①

by using backward substitution

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

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

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

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

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

$2^2[2T(n-3) - 1] - 2 - 1$

$2^3 T(n-3) - 4 - 2 - 1$

Continue for K times

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

Assume $n - K = 0 \Rightarrow n = K$

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

$2^n - 2^{n-1} - 2^{n-2} \dots 1$

$2^n - [2^{n-1} + 2^{n-2} \dots + 1]$

GP K terms

$a = 2^{n-1}$, $r = 2^{?} = \frac{1}{2}$

Sum of G.P

$a\left(\frac{1 - r^{n-1}}{1 - r}\right) = \frac{2^{n-1}(1 - (1/2)^{n-1}/1/2)}{1/2}$

$= 2^n (1 - 2(1/2)^n)$

$= \frac{2^n(2^n - 2)}{2^n} = 2^n - 2$

$T(n) = O(1)$

Q5 What should be the complexity of

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

i	s
1	1
2	3
3	6
4	10
5	15
n	K times

Ans5 $s = 1, 3, 6, 10, 15 \ldots n$

let say K terms

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

$K = 2n$ $\quad K = \sqrt{n}$ $\quad O\sqrt{n}$

t_{K-1} would be constant

Q6 Time complexity of

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

Ans6 $1^2, 2^2, 3^2, \ldots n$

let say K terms

$t_K = K^2$

$n = K^2 \Rightarrow K = \sqrt{n}$

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

Q7 Time complexity of

```
Void function (int n) {
int i, j, K, Count = 0
for (int i = n/2; i <= n; i = i+2)
    for (K=1; K<= n; K=K*2)
        Count ++;
}
```

classmate

Date
Page

Ans 7 $i = \frac{n}{2}, \frac{n}{2}+1, \frac{n}{2}+2 \cdots n$

$= \frac{n}{2}, \frac{n+2}{2}, \frac{n+4}{2} \cdots n$

given of form $= \frac{n-1\,0*2}{2} + \frac{n+1*2}{2} + \frac{n+2*2}{2} + \cdots n$

$\frac{n+k*2}{2}$ $(k=0,1,2,n)$

Total terms $= k+1$

$t_{k+1} = n$

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

$n-2 = 2k$

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

i	j	k
$n/2$	log n times	$(\log n)^2$
$\frac{n+2}{2}$	log n times	$(\log n)^2$
n	log n times	$(\log n)^2$

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

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

$T(n) = O(n \log^2 n)$

Q8 Time complexity of

function (int n) {

if (n==1) return;

for (i=1 to n) {

for (j=1 to n)

}
}
printf("S");
function(n-3);

Ans8 Function call would be n, n-3, n-6, n-9...
let says k terms
AP, $a = n$, $d = -3$
$a_n = a + (n-1)d$
$1 = n + (k-1)(-3)$
$1 = n - 3k + 3$
$3k = n + 2$
$k = \frac{n+2}{2}$

Function have recursive call $\frac{n+2}{2}$ times.

Time complexity for two for loop.
$\left(\frac{n+2}{2}\right) n^2 \Rightarrow n^3$
$T(n) = O(n^3)$

Q9 Time complexity of void function (int n) {
for (i = 1 to n)
for (j = 1; j <= n; j = j+1)
print("S");
}
}

Ans9 i (Outer loop)
when $i = 1 \rightarrow j = 1, 2, 3 \cdots n = n$
while $i = 2 \Rightarrow j = 1, 2, 3 \cdots n/2$

$\sum_{j=n}^{1}$ $n + \frac{n}{2} + \frac{n}{3} + \cdots 1$

$\sum_{j=n}^{1}$ $n\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right)$

$O(n \log n)$

Q10 For the function, n^n K ac C^{n^n} what is the asymptotic relationship b/w these function. Assume that $K \geq 1$ and $C > 1$ are constants. Find out the value of C and no for which relation hold.

Ans10 As given n^k and c^n
relation b/w n^k and c^n is $n^k = O(c^n)$

as $n^k \leq a c^n$ $\forall n \geq n_0$ for d constant $a > 0$

for $n = 1$

$c = 2$

$1^k < a 2^1$

$n_0 = 1$ and $c = 2$