Aditi Agarwal
CST-27
2017448

Page No.
Date

Tutorial 1

1. What do you understand by Asymptomatic notations. Define different Asymptotic Notation with examples.

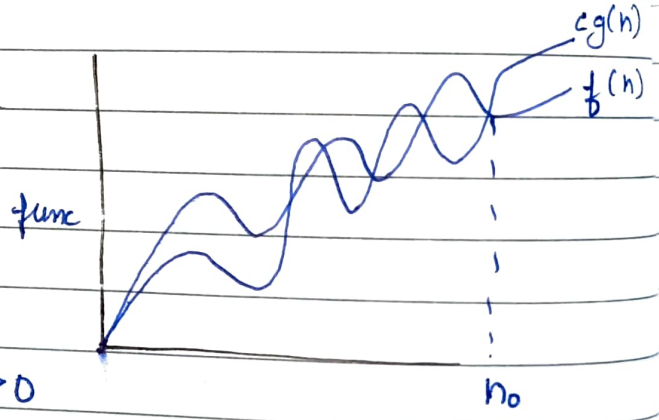
Asymptotic notation: They are the mathematical notations used to describe the running time of an algorithm where the input tends towards a particular value or a limiting value.

Different asymptotic notations -

→ **Big O(n)**

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

iff $f(n) \leq cg(n)$
$\forall n \geq n_0$



for some constant, $c > 0$
$g(n)$ is "tight" upper bound of $f(n)$

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

$n^2 + n \leq c \cdot n^3$
$n^2 + n = O(n^2)$
$n^3 = O(n^3)$

→ **Big Omega (Ω)**

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

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

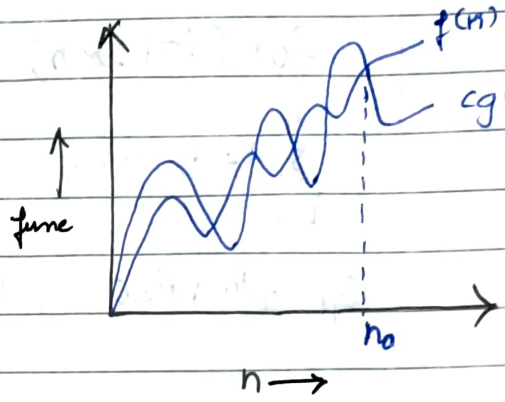
$f(n) = \Omega(g(n))$

iff

$f(n) \geq cg(n)$

$\forall \quad n \geq n_0$

for some constant $c > 0$



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

$g(n) = n^2$

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

→ **Theta (θ)**

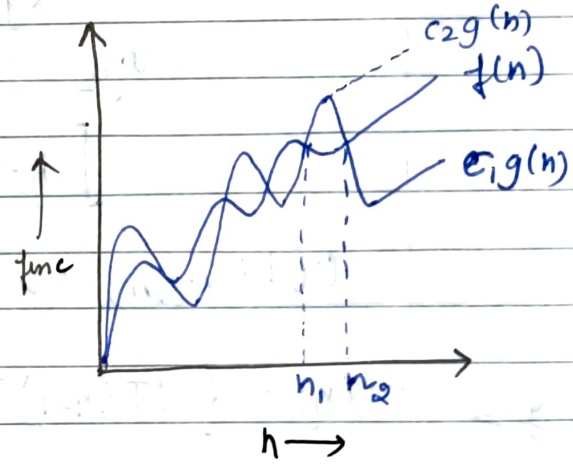
$f(n) = \theta g(n)$

$g(n)$ is both "tight" upper & lower bound of function $f(n)$

iff $c_1 g(n) \leq f(n) \leq c_2 g(n)$

$\forall \quad n \geq \max(n_1, n_2)$

where $c_1, c_2 > 0$



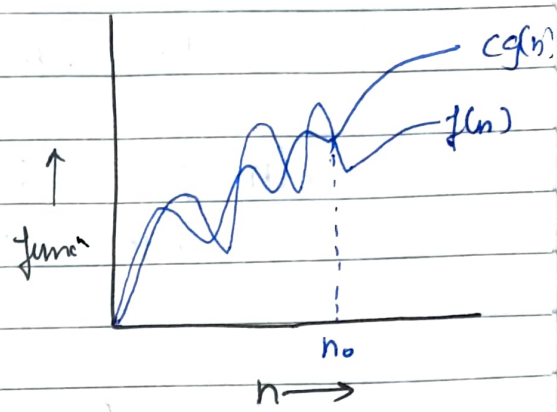
example: $3n+2 = \theta(n)$ as $3n+2 \geq 3n$ &

$3n+2 \leq 3n$ for n, $n_1 = 3$ & $n_2 = 4$ & $n_0 = 2$

→ **Small o (o)**

$f(n) = o(g(n))$

$g(n)$ is upper bound of function $f(n)$



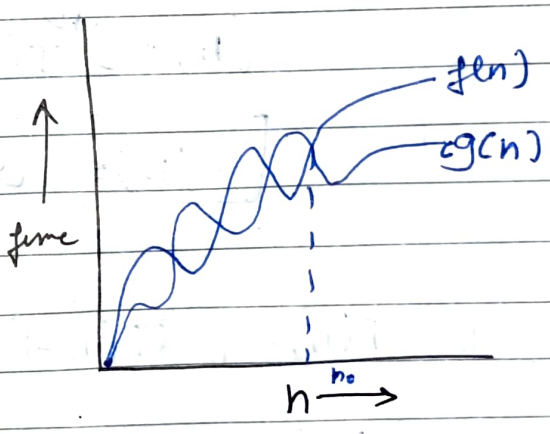
$f(n) = o(g(n))$
when $f(n) < cg(n)$

$\forall\ n > n_0$
$\exists$ $\forall$ constants, $c > 0$

ex: $f(n) = n^2$
$g(n) = n^3$
$n^2 = o(n^3)$

→ **Small omega (ω)**

$f(n) = \omega(g(n))$



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

when $f(n) > cg(n)$
$\forall\ n > n_0$

$\exists$ constant $c > 0$

2. What should be time complexity of

for (i=1 to n) { i = i*2 };

$\sum_{i=1}^{n}$ 1 + 2 + 4 + 8 ..... n

GP nth value $\Rightarrow T_k = ar^{k-1}$

$= 1 \times 2^{k-1}$

$n = 2^{k-1}$

$2n = 2^k$

$\log 2n = k \log_2 2$

$\log 2n = k$

$\log_2 2 + \log_2 n = k$

$\boxed{k = \log_2 n + 1}$

$O(1 + \log_2 n) \Rightarrow O(\log n)$

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

$T(n) = 3T(n-1)$ ——— (i)

Let $n = n-1$

$T(n-1) = 3T(n-2)$ ——— (ii)

Put 2 in (i)

$T(n) = 3 \times 3T(n-2)$ —— (3)

Put $n = n-2$

$T(n-2) = 3T(n-3)$ —— (4)

Put eqn (4) in (3)

$T(n) = 3 \times 3 \times 3T(n-3)$

$T(n) = 3^k T(n-k)$

Putting $n-k=0$

$T(n) = 3^n T(n-n) = 3^n T(0) = 3^n$

$\boxed{T(n) = O(3^n)}$

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

$T(n) = 2T(n-1) - 1$ —— (i)

Let $n = n-1$

$T(n-1) = 2T(n-2) - 1$ —— (ii)

Put (ii) in (i)

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

$= 4T(n-2) - 3$ —— (3)

Page No.
Date

Putting $n=n-2$

$$T(n-2) = 2T(n-2) - 1 \quad \text{---------- (4)}$$

Putting (4) in (3)

$$T(n) = 4[2T(n-2)-1] - 2 - 1$$
$$T(n) = 8T(n-3) - 4 - 2 - 1$$

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

GP: $2^{k-1} + 2^{k-2} + \cdots 1$

$a = 2^{k-1}$

$r = \frac{1}{2}$

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

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

Let $n-k=0$

$n=k$

$$T(n) = 2^n T(n-n) - (2^n - 1)$$
$$= 2^n T(0) - (2^n - 1)$$
$$= 2^n - 2^n + 1$$

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

5. What should be the time 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 ..... n

sum of s = $1+3+6+10 \cdots \sqrt{n}$ —— (i)

also s = $1+3+6+10 \cdots \sqrt{n-1} + \sqrt{n}$ — (ii)

(i) + (ii) ,

$0 = 1+2+3+4 \cdots n-\sqrt{n}$

$T(k) = 1+2+3 \cdots k$

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

for k iterations

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

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

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

$O(k^2) <= n$

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

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

Q.6 Time complexity of :-

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

$i^2 <= n$, $i <= n$

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

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

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

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

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

Q.7 Time complexity of :

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

⇒

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

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

Q.8 Time complexity of :

```
function (int n) {
    if (n==1) return;
    for (i=1 to n)
        for (j=1 to n)
            printf ("*");
    }
    function (n-3);          ——— T(n-3)
}
```

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

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

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

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

⋮

$T(1) = 1$

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 $= k+1$

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

$T(n) \approx n^2 + n^2 + n^2 \cdots (k+1)$ times

$T(n) \approx kn^2$

$T(n) \approx \frac{(n-1)}{3} n^2$

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

Q.9 Time complexity of :-

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

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

$i = 2$ $\quad j = 1 + 3 + 5 \cdots$

$i = 3$ $\quad j = 1 + 4 + 7 \cdots$

⋮

m^{th} term of AP is

$$T(m) = a + d \times m$$
$$= 1 + d \times m$$
$$(n-1)/d = m$$

for $i=1$ $\quad (n-1)/1$ times
$i=2$ $\quad (n-1)/2$ times
$i=3$ $\quad (n-1)/3$ times
$i=n-1$ $\quad 1$

we get,

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

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

$$= n + \frac{n}{2} + \frac{n}{3} \cdots n \times 1$$

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

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

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

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

Page No.
Date

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 & n, for which relation holds.

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

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

$\forall\ n \geq n_0$ &

constant, $a > 0$

for $n_0 = 1$

$C = 2$

$1^k \leq a^2$

$n_0 = 1$ $C = 2$.