Name :→ Harshita Mishra
University Roll no. :→ 2017498
Class Roll no. :→ 08

Semester :→ IV
Sec :→ CST-SPL-1
Date :→ 10.03.2022

TUTORIAL- 01

Ques 1 :→ What do you understand by Asymptotic notations. Define different Asymptotic notation with examples.

Ans 1 :→ **ASYMPTOTIC NOTATION :→** means Towards Infinity. These notations are used to tell the Complexity of an algorithm when the input is very Large.

Different Types of Asymptotic Notation :→

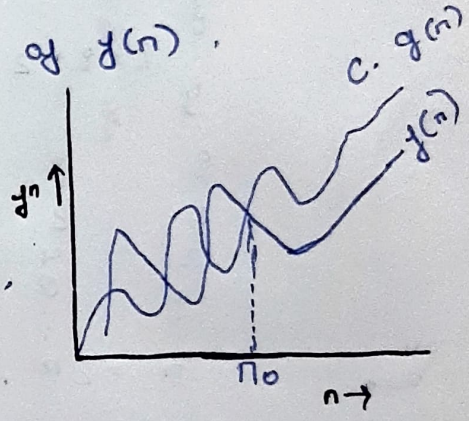
① **Big-oh (O) :→** $f(n) = Og(n)$

$g(n)$ is "tight" upper bound of $f(n)$.

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

iff. $f(n) \leq c \cdot g(n)$

$\forall\ n > n_0$, Some Const. $c > 0$,



② **Big Omega (Ω) :→**

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

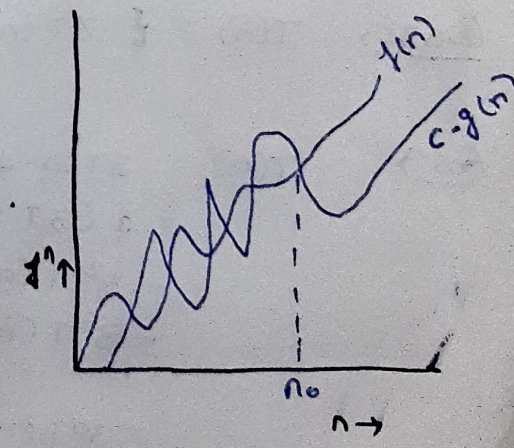
$g(n)$ is "tight" Lower bound of $f(n)$.

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

iff ; $f(n) \geq c \cdot g(n)$

~~for~~ $\forall\ n > n_0$ &

Some constant, $c > 0$.



③ Theta (Θ) :→

Theta gives the tight upper & Lower bound both.

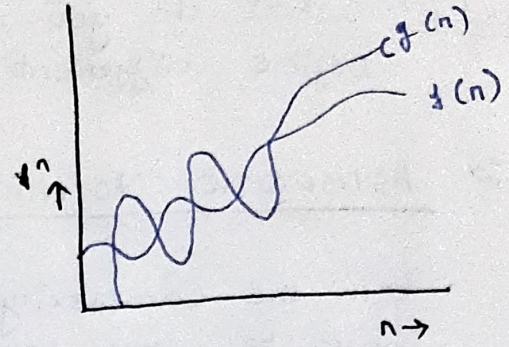
$f(n) = \Theta\, g(n)$

iff :-

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

$\forall\, n > \max(n_1, n_2)$

Some constant, c_1 & $c_2 > 0$.



Ques 2 :→ what should be time complexity of -

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

Ans :→

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

i	i
1	2
2	4
4	8
:	:
n	$= 2^k$

$\log n = \log 2^k$ → $O(\log n)$

$k = \log_2 n$ Ans

Ques 3 :→ $T(n) = \{\ 3T(n-1)$ if $n > 0$, otherwise $1\ \}$

Ans :→

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

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

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

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

$= \cdots$

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

$= 3^n T(0)$

$= 3^n \Rightarrow O(3^n)$ Ans

Ques 4:→ $T(n) = \begin{cases} 2T(n-1)-1 & \text{if } n>0, \text{ otherwise } 1 \end{cases}$

Ans:→ $T(n) = 2T(n-1) - 1$

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

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

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

$= 2^3 T(n-3) - 2^2 - 2^1 - 2^0$

- - - - - -

$= 2^n T(n-n) - 2^{n-1} - 2^{n-2} - 2^{n-3} \cdots 2^2 - 2^1 - 2^0$

$= 2^n - 2^{n-1} - 2^{n-2} - 2^{n-3} - - - 2^2 - 2^1 - 2^0$

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

$T(n) = 1$

$\Rightarrow \boxed{O(1)}$ Ans

Ques 5:→ Time Complexity of :-

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

Ans:→

i	s
1	1
2	3
3	6
4	

$1 + 3 + 6 + 10 \cdots + K = n$

$= \frac{K(K+1)(K+2)}{6} = n$

$O(K^3) = n$

$K = \sqrt[3]{n}$

$\Rightarrow \boxed{O(\sqrt{n})}$ Ans

Ques 6.:→ Time Complexity of :→

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

Ans:→ as $i^2 <= n$

$\Rightarrow i <= \sqrt{n}$

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

$\sum_{i=1}^{n} 1+2+3+4+\cdots+\sqrt{n} \Rightarrow T(n) = \frac{\sqrt{n} \times (\sqrt{n}+1)}{2}$

$T(n) = \frac{n \times \sqrt{n}}{2} \Rightarrow \boxed{T(n) = O(n)}$ Ans

Ques 7 :→ Time Complexity of :→

```
void function(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++;
}
```

Ans →

for (i = n/2; i<=n; i++) = $O(n/2) = O(n)$

for (j = 1; j<=n; j = j*2) = $k = \log_2^n = O(\log_2 n)$

for (k = 1; k<=n; k = k*2) = $x = \log_2^n = O(\log_2^n)$

$T(n) = O(n) \times O(\log_2^n) \times O(\log n)$

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

$= O(n \cdot \log^2 n) = \boxed{O(n \log^2 n)}$ Ans

Ques 8 :→ Time Complexity of :→

```
function(int n) {
    if (n==1) return;          // O(1)
    for(i=1 to n) {            // i= 1,2,3,4 ... n  = O(n)
        for (j=1 to n) {       // j=1,2,3,4 ... n^2 => O(n^2)
            printf("* ");
        }
    }
    function(n-3);             → T(n/3)
}
```

Ans :→ $T(n) = T(n/3) + n^2$

$a=1,\ b=3,\ f(n)=n^2$

$c = \log_3 1 = 0$

$\Rightarrow n^0 = 1 \quad > (T_f(n) = n^2)$

$\Rightarrow \boxed{T(n) = O(n^2)}$ Ans

Ques 9 :→ Time complexity of :→

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

Ans :→

for $i=1 \Rightarrow j = 1,2,3,4 \cdots n \quad = n$

for $i=2 \Rightarrow j = 1,3,5 \cdots n \quad = n/2$

for $i=3 \Rightarrow j = 1,4,7, \cdots n \quad = n/3$

for $i=n \Rightarrow j = 1 \cdots$

$\Rightarrow \sum_{j=n}^{1} n + n/2 + n/3 + n/4 + \cdots + 1$

$\Rightarrow \sum_{j=n}^{1} n [1 + 1/2 + 1/3 + 1/4 + \cdots \to 1/n] \Rightarrow \sum_{j=n}^{1} n(\log n)$

$T(n) = [n \log n]$

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

Question :→ For the functions, n^k and c^n, what is the asymptotic relation between these functions? assume that $k >= 1$, & $c > 1$ are constant. Find out the value of c & no for which relation holds.

Ans :→ as given, n^k and c^n

relation b/w n^k & c^n is →

$n^k = O(c^n)$ as, $n^k \leq a\,c^n$

$\forall\ n >= n_0$ and some constant areas $a > 0$

for, $n_0 = 1$

$c = 2$

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

$\Rightarrow$ **$n_0 = 1$ and $c = 2$** An