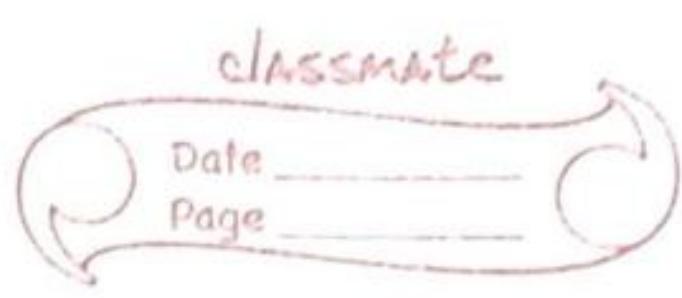


# Assignment-01



Q: 1

Solution:-

Asymptotic notations are the mathematical notations used to describe the running time of an algorithm when the input tends towards a particular value.

There are mainly 3 asymptotic notations:

Big-O notation.

Omega notation

Theta notation

- Big O( $\mathfrak{o}$ )

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

$$\text{if } f(n) \leq c_1 g(n)$$

$\forall n \geq n_0$

for some constant  $c > 0$

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

- Big Omega( $\Omega$ ):

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

$$\text{if } f(n) \geq c_2 g(n)$$

$\forall n \geq n_0$

for some constant  $c > 0$ .

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

Big Theta  $\Theta$

$$\text{if } f(n) = \Theta(g(n))$$

$$C_1 g(n) \leq f(n) \leq C_2 g(n)$$

$$+ n \geq \max(C_1, C_2)$$

for some constant  $C_1 > 0$  &  $C_2 > 0$ .

$g(n)$  is both "high upper + lower bound" of  $f(n)$ :

Q.2. Solution.

values of 'i' in 'for' loop.

$$1, 2, 4, 2^4, \dots, 2^k$$

The loop will terminate when:

$$i \geq n$$

$$2^k \geq n$$

$$k \geq \log_2(n)$$

so  $\log_2(n)$  is total unit of taken by the loop

Hence Time complexity =  $O(\log_2(n))$

Q.3.

Solution

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

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

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

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

-----

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

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

$$= 3^n$$

 $\Theta(3^n)$  Ans.

Q.4.

Solution

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

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

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

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

-----

-----

$$= 2^n T(n-n) + 2^{n-1} + 2^{n-2} + 2^{n-3}$$

-----

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

$$= T(n) = 1$$

Time Complexity is  $O(1)$

Q.5.

## Solution

Let the following loop execute  $t$  times.  
 The loop will executed till  $s$  is less than or equal to  $n$ .

$\therefore$  after first iteration -

$$s = s + 1$$

after second iteration -

$$s = s + 1 + 2$$

Since the loop iterates for the  $t$  times, we obtain.

$$\begin{aligned}s(t) &= 1 + 2 + 3 + \dots + t \leq n \\ &= O(t^2) \propto n \\ &= \sqrt{n}\end{aligned}$$

Hence complexity is  $O(\sqrt{n})$

Q.6.

## Solution

$$O(1 + \sqrt{n} + \sqrt{n} + \sqrt{n})$$

$$O(1 + 3\sqrt{n})$$

$$O(3\sqrt{n})$$

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

~~Hence complexity  $O(\sqrt{n})$~~

Q. 7

Solution

for ( $i=n/2$  ;  $i \leq n$  ;  $i++$ ) — will iterate  $n/2$  times

for ( $j=1$  ;  $j \leq n$  ;  $j^2$ ) —  $\log n$  times

for ( $k=1$  ;  $k \leq n$  ;  $k=k^2$ ) — This loop will execute  $\log n$  times  
 $\text{count}++;$

$\therefore \text{Complexity} = O(n \log^2 n)$

Q. 8

Solution

$n$	$i$	$j$
$n$	$1$	$1$
$2$	$n$	
$1$	$1$	
$(n-3)$	$n$	$n$
	$1$	$1$

 $1+4+7+\dots+n$ 

$$n = 1+3(k+1) \\ = 3k+2$$

$$k = \frac{n+2}{3}$$

↓  
No. of terms

$$\frac{n+2}{6} \left[ 2 + \left( \frac{n-1}{3} \right) \times 3 \right]$$

$$O \left[ \frac{n^2+3n+2}{6} \times n^2 \right]$$

$$O[n^4] \text{ Ans.}$$

Q. 9.

Solution

void function (int n)  
{

    for (i=1 to n) → This loop will execute n times

    {  
        for (j=1; j<=n : j=j+1) → This loop will execute  
        {  
            n/i

        } printf ("\*"); for each value of i

}

$$\therefore \text{Complexity} = n \times \sum_{i=1}^n \left( \frac{n}{i} \right), \\ = O(n \log n)$$