Tutorial - 1

Date.
Page No.

Q1 Define Asymptotic Notations with examples.

The efficiency of an algorithm depends on the amount of time, storage and other resources required to execute the algorithm. The efficiency is measured with the help of asymptotic notations.

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

For example: In bubble sort, when the input array is already sorted, the time taken by the algorithm is linear (i.e the best case).

But when the input array is in reverse condition, the algorithm takes the maximum time (quadratic) to sort the elements i.e the worst case.

$\log 2^k$ $\quad 2^k + 1 = n$ $\quad$ Date. $2^k = n$

$2^k + 1$ $\quad 2$ $\quad$ Page No.

Q2 $\quad$ for(int i=1 to n) (i = i × 2);

kth step

$$\boxed{2^k = n}$$

taking log both time

$$\log 2^k = \log n$$

$$k \log_2 2 = \log n$$

$$\boxed{k = \log n}$$

Q3) $\cancel{T(n) = \{ 3T(n-1) \text{ if } n > 0, \text{ otherwise } 1}$

$$\boxed{\text{Time Complexity} \Rightarrow O(\log n)}$$

Date.
Page No.

Q3. $T(n) = \{3T(n-1)$ if $n > 0$, otherwise $1\}$

$$T_n = \begin{cases} 1 & n \le 0 \\ 3T(n-1) & n > 0 \end{cases}$$

$(T(n) = 3T(n-1)$ — ①

Let $n = n-1$

$T(n-1) = 3T(n-2)$ — ②

Let $n = n-2$

$T(n-2) = 3T(n-3)$ — ③

Substitute ② in ①

$T(n) = \cancel{3T(n}\, 3(3T(n-2))$

$T(n) = 9T(n-2)$ — ④

Substituting ③ in ④

$T(n) = 9(3T(n-3))$

$T(n) = 27T(n-3)$ — ⑤

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

Date.
Page No.

for $n-k=0$

$$\boxed{n=k}$$

$$T(0)=1$$

$$T(n)=3^k\,T(0)$$

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

Put $k=n$

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

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

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

$$T(n) = \begin{cases} 1 & n = 0 \\ 2T(n-1) - 1 & n > 0 \end{cases}$$

$T(n) = 2T(n-1) - 1$ — (1)

Let $n = n-1$

$T(n-1) = 2T(n-2) - 1$ — (2)

Let $n = n-2$

$T(n-2) = 2T(n-3) - 1$ — (3)

$\Rightarrow$ (2) in (1)

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

$T(n) = 4T(n-2) - 2 - 1$

$T(n) = 4T(n-2) - 3$ — (4)

$\Rightarrow$ (3) in (4)

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

$T(n) = 8T(n-3) - 4 - 3$

$T(n) = 8T(n-3) - 7$ — (5)

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

$$n - k = 0$$

$$\boxed{n = k}$$

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

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

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

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

~~Time Complexity = $O(2^n)$~~

$$T(n) = 1$$

Time Complexity $\Rightarrow O(1)$

5. int i = 1, S = 1;

```
While ( S <= n) {        // O(n)
   i++ ; S = S + i;      // O(1)
   printf ("#");
}
```

i i i i

1 2 3 4 5 6 7 8 9 10

i	S
1	1
2	3
3	6
4	10
5	15

1+2

$$K \qquad \frac{K(K+1)}{2}$$

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

$\cancel{K^2 + K = 2n}$

$\cancel{K^2 = 2n}$

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

Q6:

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

$i \Rightarrow \boxed{1, 4, 9, 16 \ldots (\sqrt{n})^2} = K$

$AK^2 + BK + C = tK$

at $K=1$ $\quad A+B+C = 1$ — ①

at $K=2$ $\quad 4A+2B+C = 4$ — ②

at $K=3$ $\quad 9A+3B+C = 9$ — ③

From ①, ② & ③

$A=1, \quad B=0, \quad C=0$

$n = AK^2 + BK + C$

$n = AK^2 + 0 + 0$

$n = K^2$

$K = \sqrt{n}$

Time Complexity $= O(n^2)$

Date.
Page No.

Q7

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

Loop 1 // $O(n/2)$

Loop 2 // $O(\log n)$

Loop 3 // $O(\log n)$

Time Complexity = $\frac{n}{2} \times \log n \times \log n$

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

Q8 function (int n)
{
if (n==1) return;
for(i=1 to n)
{
for (j=1 to n)
{
printf ("*");
}}
function (n-3); }

$(n-3), (n-6), (n-9) \ldots (1)$

$a = n-3,\ d = \cancel{n}-6-\cancel{n}+3 \ldots$

$1 = (n-3) + (k-1)(-3)$

$1 = (n-\cancel{3}) - 3k + \cancel{3}$

$3k = n-1$

$k = \frac{n-1}{3} = O(n \times n^2)$

$\underline{O(n^3)}$

Q9

Date.
Page No.

Q9 void function (int n)

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

for $i=1$, $j=n$ times

$i=2$ $\quad j=n/2$ times

$\vdots$ $\quad \vdots$

$i=4$ $\quad j=n/4$ times

$i=n$ $\quad j=1$

Time Complexity $= n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} \dots \frac{n}{n}$

$\Rightarrow n\underbrace{\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \dots 1\right)}_{\log n}$

$\Rightarrow$ ~~O (n log n)~~ $O(n \log n)$