Date

tutorial 1

Ans-1) Asymptotic notations are mathematical notations which are used to tell the complexity of an algorithm when the input is very large.

types of asymptotic notations:-

1) Big-Oh (O) → It gives "tight" upper bound.

2) Big-Ohmega (Ω) → It gives "tight" lower bound

3) theta (θ) → It give "tight" upper and "tight" lower bound both.

4) small-Oh (o) → It gives upper bound.

5) small ohmega (ω) → It gives lower bound.

Ans-2) $O(\log_2 n)$

Ans-3) $T(n) = 3T(n-1) \quad n > 0$

$T(1) = 1 \quad n <= 0$

By forward substitution :-

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

$T(0) = 1$

$T(1) = 3T(1-1) = 3T(0) = 3$

$T(2) = 3T(2-1) = 3T(1) = 9 \Rightarrow 3^2$

$T(3) = 3T(3-1) = 3T(2) = 3 \cdot 3^2 = 3^3$

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

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

Ans-4) $T(n) = \begin{cases} 2T(n-1) - 1 & n > 0 \\ 1 \end{cases}$

by forward substitution

$T(0) = 1$

$T(1) = 2T(1-1) - 1 \Rightarrow (2-1) = 1$

$T(2) = 2T(2-1) - 1 \Rightarrow 2T(1) - 1 = 1$

$T(3) = 2T(3-1) \Rightarrow 3T(2) - 1 = 1$

$T(n) = 1$

$\therefore \boxed{TC = O(1)}$

Date
Page

Ans-5)

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

s=3 6
i=2 3 4

#

1 → 3 → 6 → 10 → 15

$\Rightarrow (1) + (1+2) + (1+2+3) + (1+2+3+4) + \cdots$

$\Rightarrow 6n(n+1)(n+2)$

$\Rightarrow O(n^3)$

Ans-6) 1, 2, 3, 4, 5, 6, 7, 8, 9

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

Ans-7) $O\left[\left(\frac{n}{2}\right) \cdot \log_2 n \cdot \log_2 n\right] \Rightarrow O(n \log n \log n)$

Ans-8) $T(n) = T(n-3) + n^2$

$T(1) = 1$

Ans-9)

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

(i)	(j)	Values
1	$1 \to n$	n
2	$1 \to n$	$(n+1)/2$
3		
.		
.		
n		

$$TC \Rightarrow O(n) * O(n) \Rightarrow O(n^2)$$

Ans-10) n^k is $O(c^n)$.

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