

Tutorial - 2

$j=1$	$i=0$
1	1
2	3
3	6
4	10

```
for (i=0 ; i<n ; i++)
    for (j=0 ; j<n ; j++)
        for (k=0 ; k<n ; k++)
            c++;
```

```

(iii)  $\log(\log n)$ 
int func (int n)
{
    if (n==1)
        return n;
    else
        return func( $\sqrt{n}$ ) + func( $\sqrt{n}$ );
}

```

Ans 4. $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + n^2$

Using master's method :

$$a=2 \quad b=2 \quad c=1$$

$$f(n) > n^2 \quad f(n^2) > 1$$

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

Ans 5. int func (int n) {

for (int i=1 ; i<=n ; i++) {

for (int j=1 ; j<=n ; j++) {

c++;

}}

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

Ans 6. for (int i=2 ; i<=n ; i=pow(i,k))

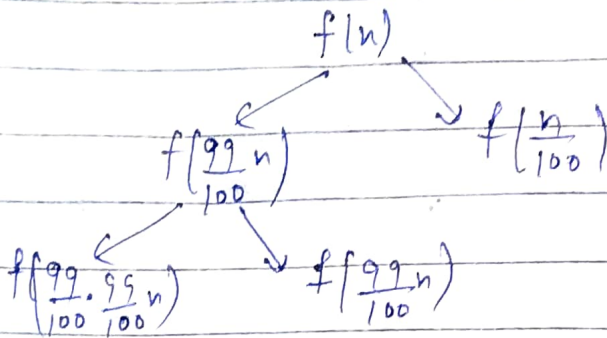
{

c++;

}

where, k is a constant.

Ans 7. $T(n) = T\left(\frac{99}{100}n\right) + T\left(\frac{n}{100}\right)$



Time complexity = $O(\log n)$

Ans 8.

a) $1 \ll \log(\log(n)) < \log(n) < \sqrt{n} < n < \log(n!) < n \log(n) < n^2 < 2^n < 2^{2n} < 4^n < n!$

b) $1 < \log(\log(n)) < \sqrt{\log n} < \log(2^n) < \log(n) < 2(\log(n)) < n \log(n) < n < 2n < 4n < n^2 < \log(n!) < 2(2^n) < n!$

c) $96 < \log_2(n) < \log_8(n) < \log_5 n < \log n! < n \log_6(n) < n \log_2(n) < 8n^2 < 7n^3 < 8^{2n} < n!$

Name - Sristi Gangwar

Section - E

Roll no. - 2014897 (56)