Date

Page No.

Tutorial - II

Ans 1)

```
void function (int n)
{
    int j=1 , i=0;
    while (i<n)
    {
        i = i+j
        j++;
    }
}
```

$j=1$, $i = 0+1$

$j=2$, $i = 0+1+2$

$j=3$, $i = 0+1+2+3$

$\vdots$

loop ends when $i >= n$

$0+1+2+3 \ldots n > n$

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

$$k^2 > n$$

$$k > \sqrt{n}$$

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

Ans 2) Recurrence Relation for Fibonacci Series:

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

$$T(0) = T(1) = 1$$

if $T(n-1) \approx T(n-2)$

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

$$= 2\{2T(n-4)\} = 4T(n-4)$$

$$= 4(2T(n-6))$$

$$\vdots$$

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

Date

Page No.

$n - 2k = 0$

$n = 2k$

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

$T(n) = 2^{n/2} \; T(0)$

$= 2^{n/2}$

$T(n) = \Omega(2^{n/2})$

if $T(n-2) \approx T(n-1)$

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

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

$= 4(2T(n-3)) = 8T(n-3)$

$= 2^k \; T(n-k)$

$n - k = 0$

$\boxed{k = n}$

$T(n) = 2^k \times T(0) = 2^n$

$= T(n) = O(2^n)$ (upper bound)

Ans 3) $O(n(\log n))$ ⇒

```
for( int i=0 ; i<n; i++)
{
    for( int j=1; j<n; j = j*2)
    {
        // some O(1)
    }
}
```

Mohan

Teacher's Signature

$O(n^3)$ =>

```
for(int i=0; i<n; i++)
{
    for(int j=0; j<n; j++)
    {
        // some O(1)
    }
}
```

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

```
for(int i=1; i<=n; i=i*2)
{
    for(int j=1; j<=n; j=j*2)
    {
        // some O(1)
    }
}
```

Ans 4) $T(n) = T(n/4) + T(n/2) + cn^2$

Let's assume

$$T(n/2) >= T(n/4)$$

So, $T(n) = 2T(n/2) + n^2$

applying Master's Theorem

$$(T(n) = aT(\tfrac{n}{b}) + f(n))$$

$a=2$, $b=2$, $f(n) = n^2$

$$c = \log_b a = \log_2 2 = 1$$

$$n^c = n$$

Compare n^c and $f(n) = n^2$

$$f(n) > n^c$$

So, $T(n) = \theta(n^2)$

Ans 5)

```
int fun (int n)
{
    for ( int i=1; i<=n; i++)
    {
        for ( int j=1; j<n; j+=i)
        {
            // some O(1)
        }
    }
}
```

$i=1$ — $j=1$, $j=2$, $j=3$, ⋮, $j=n$ — n times

$i=2$ — $j=1$, $j=3$, $j=5$, $j=7$ — Loop ends when $j>n$

$1+3+5+7 > n$

$k > \frac{n}{2}$

— n times

$i=3$ — $j=1$, $j=4$, $j=7$ — $1+4+7 > n$

$k > \frac{n}{3}$

$i=4$ — $k > \frac{n}{4}$

⋮

$i=n$

So, Total complexity = $O(n^2 + n^2 + n^2 + \ldots)$
$= O(n^2)$

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

// some (1)

}

complexity of pow (i, k) = $O(\log N)$
$= \log(k)$

$i = 2$
$i = 2^k$
$i = 2^{k^2}$
$i = 2^{k^3}$
$i = 2^{k^4}$
⋮
$i = 2^{k^m}$

Loop ends when
$i > n$
$2^{k^m} > n$
$\log(2^{k^m}) > \log n$
$k^m > \log n$
$\log(k^m) > \log(\log n)$
$M \log(k) > \log(\log n)$
$M > \frac{\log(\log n)}{\log k}$

$T(c) = O(\log(\log n))$

Ans 8)

a) $100 < \log n < \sqrt{n} < n < \log(\log n) < n \log n < \log n!$
$< n! < n^2 < \log 2^n < 2^n < 2^{2n} < 4^n$

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

c) $96 < \log_8 N < \log_2 N < N \log_6 N < n \log_6 N < n \log_2 N <$
$\log N! < N! < 5N < 8N^2 < 7N^3 < 8^{2n}$