Tutorial - 2

1.

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

Ans: i can be defined as $i_j = i_{j-1} + j$. The value of j increase for each iteration by 1. The value of i at jth iteration is the sum of first j positive integers. If total iteration is s so

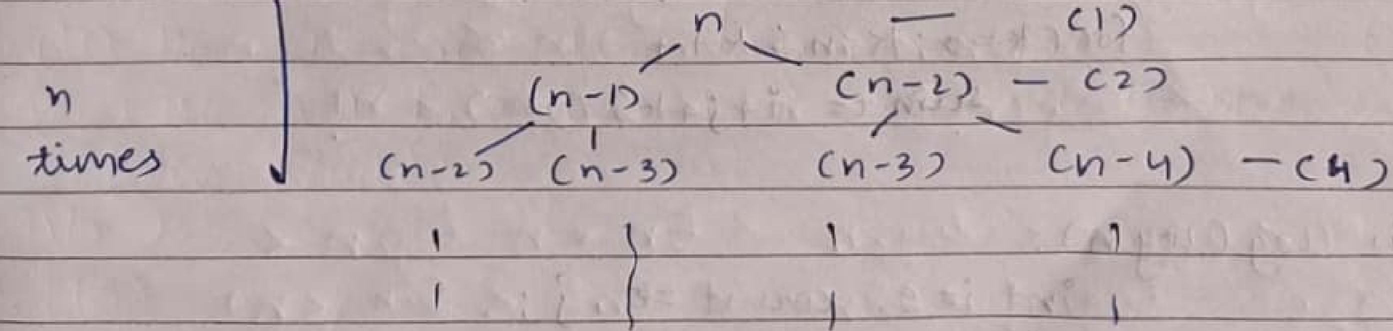
$$1+2+3+\dots+s = \left[\frac{s(s+1)}{2}\right] > n$$

so $s = O(\sqrt{n})$.

$T.C = O(\sqrt{n})$.

2. Recurrence Relation for recursive function that prints fibonacci series is

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



~~Result~~

$T = 1+2+4+\dots+2^n$

Page No.
Date : / . /

$a = 1, \quad r = \frac{2}{1} = 2.$

$$\frac{a(r^t - 1)}{(r-1)} = \frac{1(2^{n+1} - 1)}{(2-1)}$$

$$= 2^{n+1} - 1$$

$$T.C = O(2^n).$$

without recursion space complexity = O(1).
Because there are n no of function calls during recursion so S.C = O(n).

3) (i) Program having T.C = O(n log n).

```
int sum = 0, i, j;
for (i=1; i<=n; i*=2)
{  for (j=1; j<=n; j++)
      {   sum += j;
      }
}
```

(ii) T.C = $O(n^3)$.

```
int i, j, k, sum=0;
for( i=0; i<n; i++
   for( j=0; j<n; j++)
      for( k=0; k<n; k++
         sum += i+j+k;
```

(iii) log (log n).

```
int i=2, count =0, j;
j = j*2;
while (j <n) {
      count++;
      j += i;
}
```

Page No.
Date : / /

4. $T(n) = T(\frac{n}{4}) + T(\frac{n}{2}) + cn^2$

$T(\frac{n}{2}) >= T(\frac{n}{4})$

eqn can be written as

$T(n) = 2T(\frac{n}{2}) + cn^2$

using master's method.

$a=2, b=2,$

$c = \log_a b = \log_2 2 = 1.$

$n^c = n$

$F(n) = n^2$

$F(n) > n^c$

$T.C = \Theta(F(n)) = O(n^2)$

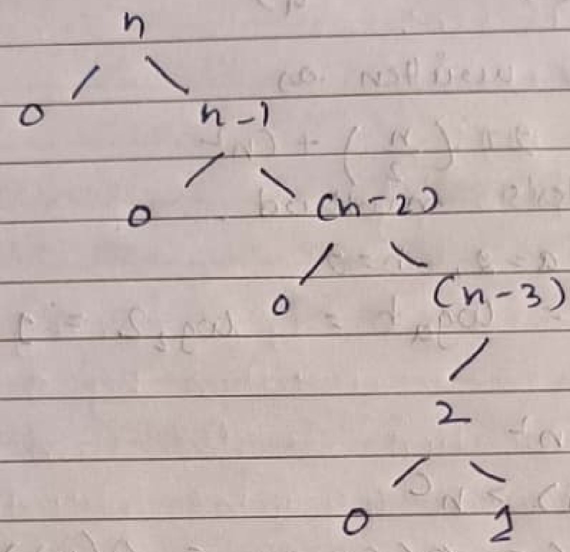
5.
```
int fun(int n) {
    for (int i = 1; i<=n; i++)
        for(int j=1; j<n; j+=i)
            // O(1) task
}
```
$T.C = O(n^2).$

6.
```
for (int i= 2; i<=n; i=pow(i,k))
    { // O(1) task ;
    }
```
k is a constant

$T.C = O(\log(\log n)).$

7. Recurrence Relation will be

$$T(n) = T\left(\frac{9n}{10}\right) + T\left(\frac{n}{10}\right) + O(n).$$



$cn + c(n-1) + c(n-2) + --- + 2c =$

$c(n + (n-1) + (n-2) + --- + 1)$

$= O(n^2).$

(8)

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

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

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