

## Tutorial-2

DATE    PAGE   Ques 1

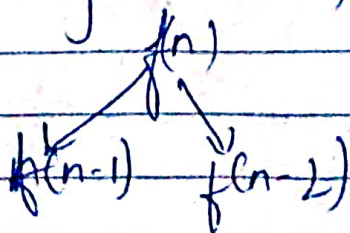
```
void fun (int n) {  
    int i = 1, j = 0;  
    while (i < n) {  
        i = i + j;  
        j++;  
    }  
}
```

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

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

Ques-2 Write recurrence relation for the recursive function that prints fibonacci series.  
Solve recurrence relation to get time complexity of the program?  
What will be the space complexity of this program and why?

Ans-2 Space complexity =  $O(n)$  as recursion calls for  $f(n-1)$   
Time complexity =  $O(2^n)$





Ques- 3 Write programs which have complexity  
-  $n(\log n)$ ,  $n^3$ ,  $\log(\log n)$

Ans  $n \log n \rightarrow$  for ( $i=0; i < n; i++$ )  
for ( $j=0; j < n; j=j+2$ )  
printf(" ");

$n^3 \rightarrow$

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

$\log(\log n)$

```
int fun(int n) {
  if (n==1)
    return n;
  else
    return fun(sqrt(n)) + fun(sqrt(n));
}
```

Ques- 4 Solve the following recurrence relation  
 $T(n) = T(n/4) + T(n/2) + (n^2)^2$

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

using master theorem

$a=2, b=2$   
 $(=1)$

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



Ques-5

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

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

Ques-6

```
for (int i = 2; i <= n; i = pow(i, k))  
    // some task  
}
```

 where  $k$  is constant

Ans - 6  $O(\log \log n)$

Ques-7  $T(n) = T\left(\frac{99}{100}n\right) + T\left(\frac{n}{100}\right)$   
 $= TC O(\log n)$

Ques-8 a)  $100 < \log \log n < \log n < \sqrt{n} < n \log(n-1) < n \log n < n^2 < 2^n < 2^{2n} < 4^n < n!$

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

c)  $96 < \log n < \log 5n < \log n! < n \log 2 < n \log 2n < 8n^2 < 7n^5 < 8^{2n} < n!$