Tutorial - 2

Ans-1

i	j
1	2
3	3
6	4
10	5

$$K^2 = n$$
$$K = \sqrt{n}$$
$$TC = O(\sqrt{n})$$

$$\frac{K(K+1)}{2} = n$$

Ans-2

$$T(n) = T(n-1) + T(n-2)$$
$$T(0) = 0, T(1) = 1$$

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

Using backward solution

$$T(n) = 2 \times 2(T(n-2)+1)+1$$
$$= 4(T(n-2)+3)$$

$$T(n-2) = 2T(n-3)+1$$
$$T(n) = 2(2(2(T(n-3)+1)+1)+1)$$
$$= 8T(n-3)+3$$

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

$\underline{n = K}$

$$T(n) = 2^n + 2^n + 1$$
$$TC = O(2^n)$$

Ans-3

O(n logn)

```
Void func(int n) {
    for( int i = 1; i<=n; i++) {
        for( int j = 1; j<=n; j=j*2) {
            Task;
        }
    }
}
```

O(n³)

```
Void func (int n) {
    for ( int i=0; i<n; i++ ) {
        for (int j=0; j<n; j++){
            for ( int k=0; k<n; k++){
                Task;
            }
        }
    }
}
```

O(log(logn))

```
Void func (int n) {
    for( int i=n; i>1; i= Pow(i, k)) {
        Task;
    }
}
```

Ans-4

$$T(n) = T(n/4) + T(n/2) + Cn^2$$

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

$$T(n) = 2T(n/2) + Cn^2$$

$$C = \log_b a$$

$$= \log_2 2 = 1$$

$$n^C < f(n)$$

$$\underline{TC = O(n^2)}$$

Ans-5

i	j
1	n times
2	$n/2$ times
3	$n/3$ times
n/n	n/n times
	$\log n$

$TC = O(n \log n)$

Ans 6

$i = 2, 2^K, 2^{K^2}, 2^{K^3} \ldots\ldots 2^{K \log_K (\log(n))}$

$2^{K \log_K (\log n)} = n$

$2^{\log(1)} = 1$

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

Ans-7

$T(n) = T(9n/10) + T(n/10) + O(n)$

taking one branch 99% and other 1%.

$T(n) = T(99n/100) + T(n/100) + O(n)$

1^{st} level $= n$

2^{nd} level $= 99n/100 + n/100 = n$

So 3^{rd} remain same for any kind of position. if we take longer branch $= O(n \log_{\frac{100}{99}} n)$. for shorter branch $= \Omega(n \log_{10} n)$ either way base complexity of $O(n \log n)$ remain.

Ans-8

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

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

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