Name- Abhishek Singh

Semester- III

Section - CST-SPL-02

Class Roll no.- 09

~~Eg~~ Tutorial- 02

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

Initially :-> i=0 , j=1

	value at i	value at j	Run
1st time	i=1	j=2	1
2nd time	i=3	j=3	1
3rd time	i=6	j=4	1
4th time	i=10	j=9	1
5th time	i=15	j=6	1
6th time	i=21	j=7	1

kth time $i = n$ $j = k$ 1

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

$$k^2 + k = 2n$$

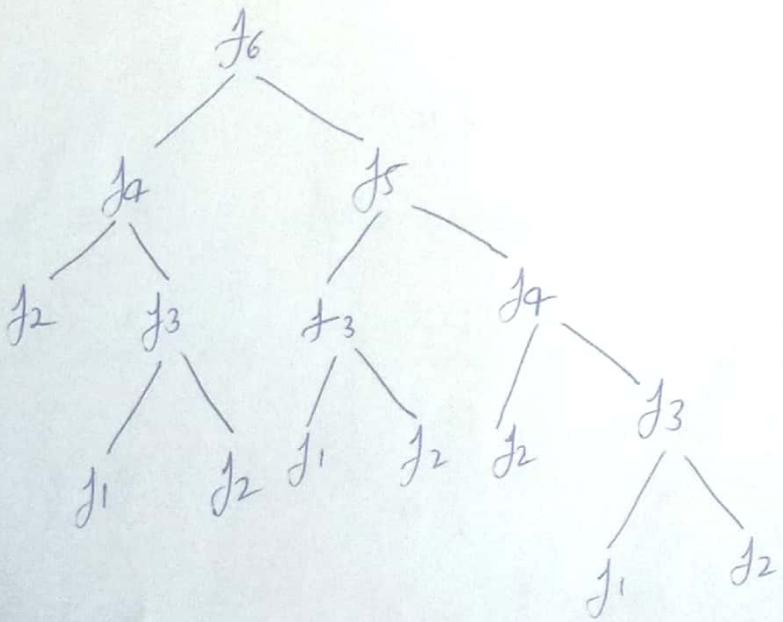
$$k^2 \leq n$$

$$k = \text{sqrt}(n)$$

time complexity → $O(\sqrt{n})$

Ans. 2>

```
int fib (n)
{
    if(n== 0 || n ==1)
        return 1;
    else
        return fib(n-1) + fib(n-2);
}
```



$T = O(2^n)$

space complexity = $O(N)$

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

$\Rightarrow T(n-1) \approx T(n-2) = 2T(n-2)$

$T(n-2) = 2*(2T(n-2)) + c$

$= 2^2 T(n-2) + c$

$T(n-4) = 2*(4T(n-2) + 3c) + c$

$= 8T(n-3) + 7c$

$\Rightarrow 2^k(T-k) + (2^k - 1)c$

let, $n-k = 0$

$n = k$

$= 2^k + (0) + (2^n - 1)c$

~~$2^n + (0) + 2$~~

$2^n \cdot 1 + 2^n c - c$

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

$\approx 2^n$

Ans3) $n(\log n)$, n^3 , $\log(\log n)$

1) we can use 3 loops -> $O(n^3)$

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

ii) for time complexity -> log (log n)

```
for(int i=2 ; i<n ; i = pow(i, 2))
{
        // O(1)
}
```

iii) for time complexity => n log n

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

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

$T(n/2) \geq T(n/4)$

$\Rightarrow T(n) = 2T(n/2) + cn^2$

Using master method we get

$b=2, \quad a=2$

$c = \log_b a$

$= \log_2 2$

$\Rightarrow c = 1$

$f(n) > n^c$

$T(n) = O(f(n))$

$= O(n^2)$

Ans.5> int fun (int n) {

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

~~for j~~

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

//O(1)

}

}

}

So, $T(n) = n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + ----$

$= n\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots\right)$

$= n \int_1^n \frac{1}{x}$

$= n [\log x]_1^n$

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

Ans.6> for first iteration $i = 2$

2nd iteration $i = 2^k$

3rd iteration $i = (2^k)^k$

⋮

n^{th} iteration $\Rightarrow i = (2^k)^i$

loop ends at $2^{k^i} = n$

$\Rightarrow \log n = \log 2^{k^i}$

$\Rightarrow k^i = \log n$

$\Rightarrow \log k^i = \log(\log n)$

$\Rightarrow i = \log(\log n)$

$T(n) = \Theta(\log(\log(n))$

Ans 3)

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

b) $1 < \sqrt{\log n} < \log_2 n = 2\log n = \log n < 2n = 4n = n$
$< \log(n!) < n\log n < n^2 < 2(2^n) < n!$

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