Tutorial-2

Sol-1:)

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

i	j
0	1
1	2
3	3
6	4
10	5
15	6
	7

Series : 0, 1, 3, 6, 10, 15 ----

$$n = 0 + 1 + 2 + 3 + \cdots + k$$

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

$$n = \frac{k^2 + 1}{2}$$

$$n \cong k^2$$

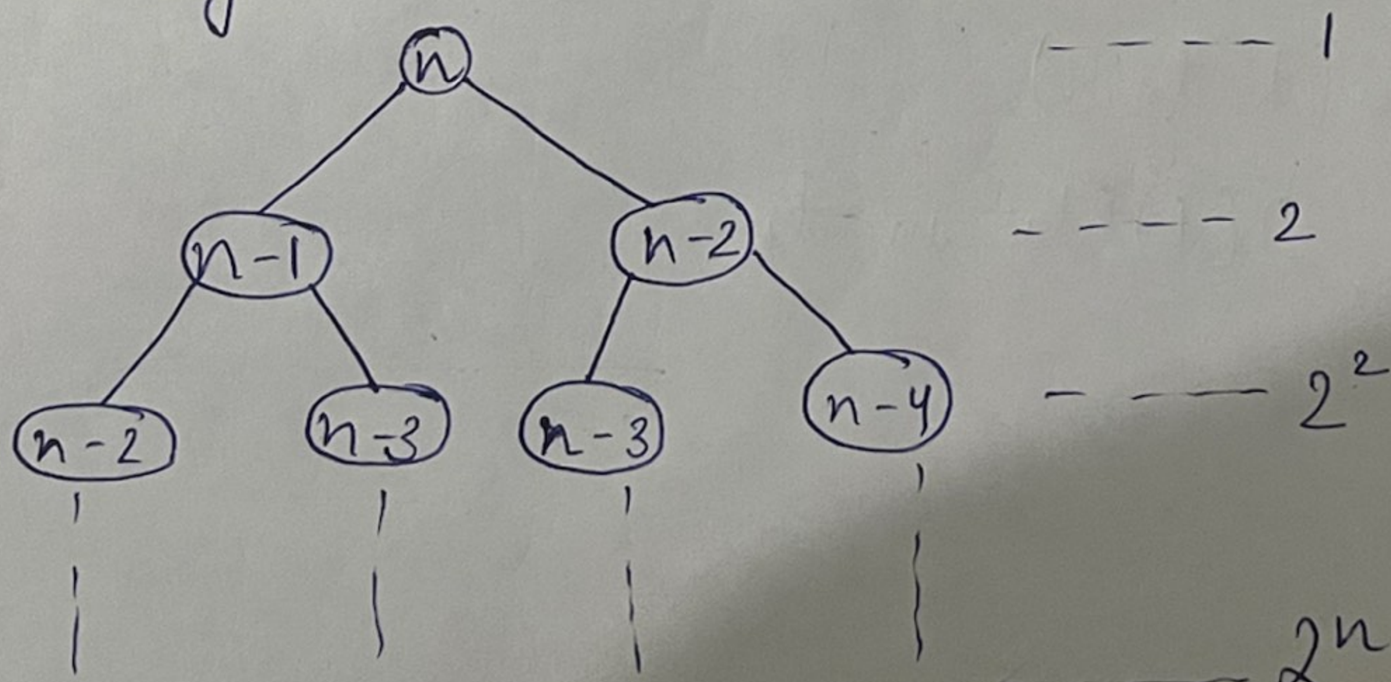
$$k \cong \sqrt{n}$$

$$\boxed{TC = O(\sqrt{n})}$$

Sol-2:) Recurrence relation for fibonacci series

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

Using Recurrence tree method



$$T.C. = 1 + 2 + 4 + \cdots + 2^n$$

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

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

Space complexity: Space complexity of fibonacci series using recursion is proportional to height of recurrence tree.

So, $\boxed{SC = O(n)}$

Sol-3:)(i) $n \log n$

```
for (i to n)
{
   for (j=1 ; j<=n ; j*=2)
              O(1) statements
}
```

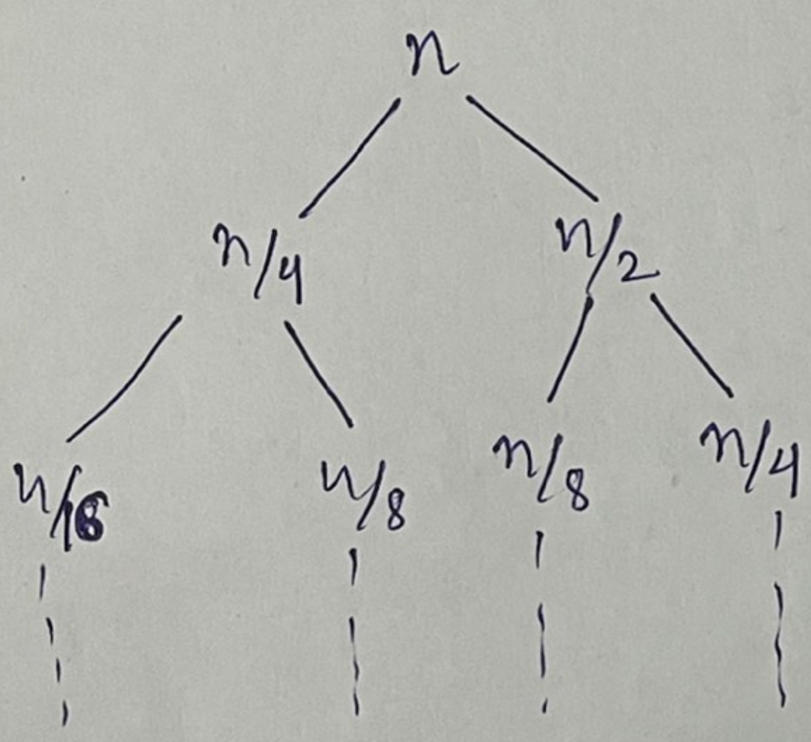
(iii) n^3

```
for (i to n)
   for (j to n)
      for (k to n)
         O(1) Statements
```

(iii) $\log(\log n)$

```
int i = n;
while (i > 0)
{
    i = √i;
}
```

Sol-4:) $T(n) = T\{n/4\} + T(n/2) + Cn^2$



$\rightarrow Cn^2$

$\rightarrow \frac{Cn^2}{16} + \frac{Cn^2}{4} = \frac{5Cn^2}{16}$

$\rightarrow \frac{Cn^2}{256} + \frac{Cn^2}{64} + \frac{Cn^2}{64} + \frac{Cn^2}{16} = \frac{25Cn^2}{256}$

So, $T(n) = C\left(n^2 + \frac{5n^2}{16} + \frac{25n^2}{256} + \text{----}\right)$

here, $r = \frac{5}{16}$ So, $S_n = \frac{1}{1-r}$

$$T(n) = Cn^2\left(1 + \frac{5}{16} + \frac{25}{256} + \text{---}\right)$$

$$= Cn^2\left(\frac{1}{1 - 5/16}\right)$$

$$= Cn^2 \cdot \frac{16}{11}$$

$\therefore TC = \Theta(n^2)$

Sol-5:)
```
int for (int n){
    for (int i = 1; i <= n ; i++){
        for (int j = 1 ; j < n ; j += i) {
            // same O(1) task
        }
    }
}
```

i	j	time
1	1 to n	(n-1)
2	1 to n	(n-1)/2
3	1 to n	(n-1)/3
⋮	⋮	⋮
n	1 → n	(n-1)/n

$n \log n$

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

Sol-6:)
```
for (int i = 2; i <= n; i = pow(i, k)){
    {
        // same O(1) expressions or standards
    }
}
```

$i = 2, 2^k, 2^{k^2}, 2^{k^3} \dots\dots 2^{k^n}$

$n = 2^{k^n}$

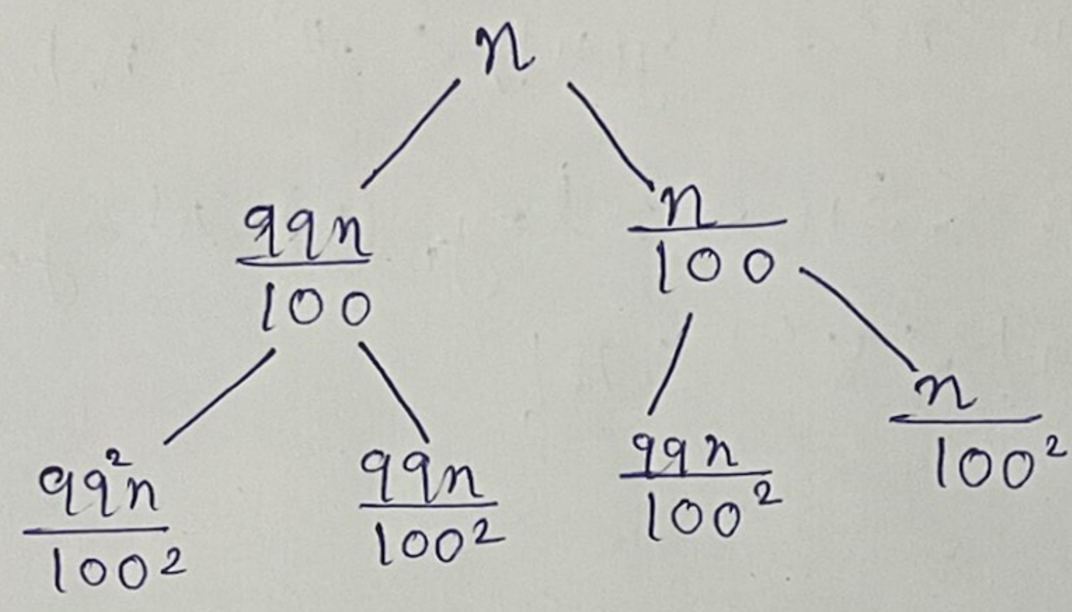
$\log n = k^n \log 2$

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

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

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

Sol-7:)



Taking longer branch i.e. $\frac{99n}{100}$

$$TC = \log \frac{100n}{99}$$

$$\cong \log n$$

$$n = \left(\frac{99}{100}\right)^K$$

$$\text{or } K = \log\left(\frac{100n}{99}\right)$$

$$T(n) = n\left(\log \frac{100}{99}\right)^n / 100$$

$$= O(n \log_{99} n)$$

Sol-8:) Increasing order of rate of growth.

(a) $n,\ n!,\ \log n,\ \log \log n,\ \text{root}(n),\ \log(n!),\ n \log n,$
$\log^2 n,\ \log 2^n,\ 2^{2^n},\ 4^n,\ n^2,\ 100$

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

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

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