Q.11 ⇒

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

Series ⇒ 0, 1, 3, 6, 10, 15 --

let at last iteration :

$n = 0 + 1+2+3+4+5+ \dots + k$

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

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

$n \cong k^2$

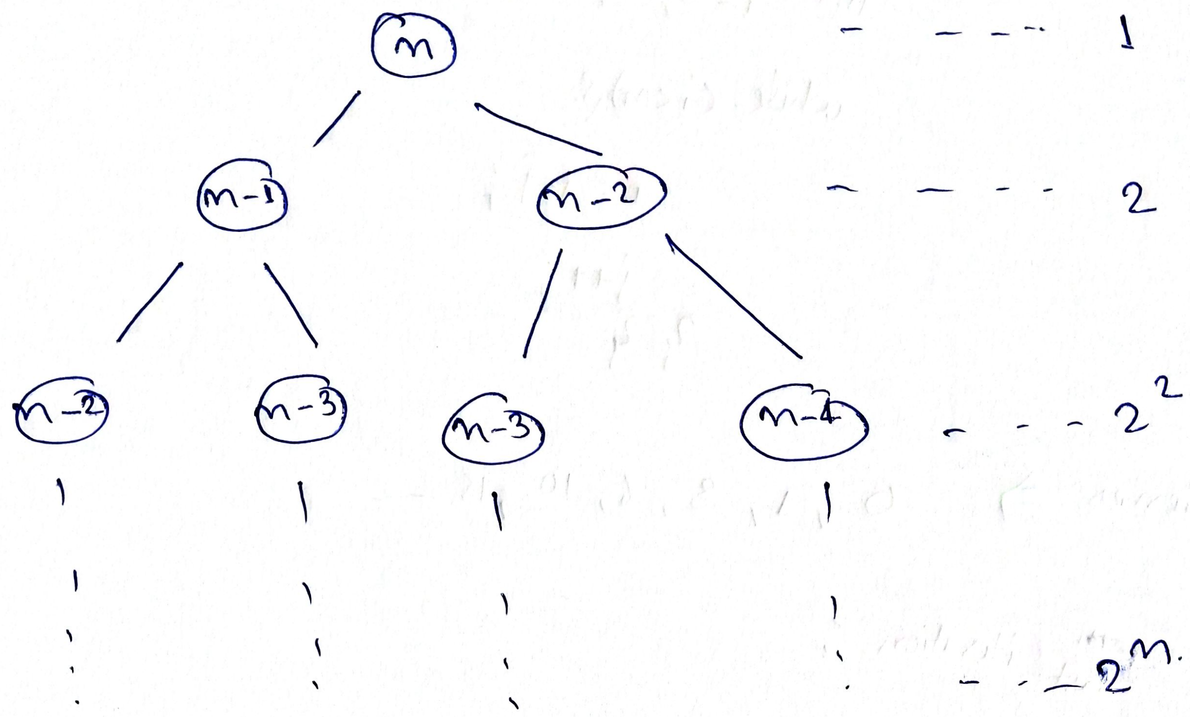
$k \cong \sqrt{n}$

so T.C. ⇒ $O(\sqrt{n})$

Q.12 ⇒ Recurrence relation for fibonacci series.

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

using Recurrence tree method :



$$T.C. = 1 + 2 + 4 + \cdots + 2^n = 1\,\frac{(2^{n+1}-1)}{2-1} = 2^{n+1} - 1$$

so $T.C. = O(\cancel{n}\,2^n)$

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

so $S.C. \rightarrow O(n)$

Q.13 ⇒ Write code for complexity.

(i) $n \log n$

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

(ii) n^3

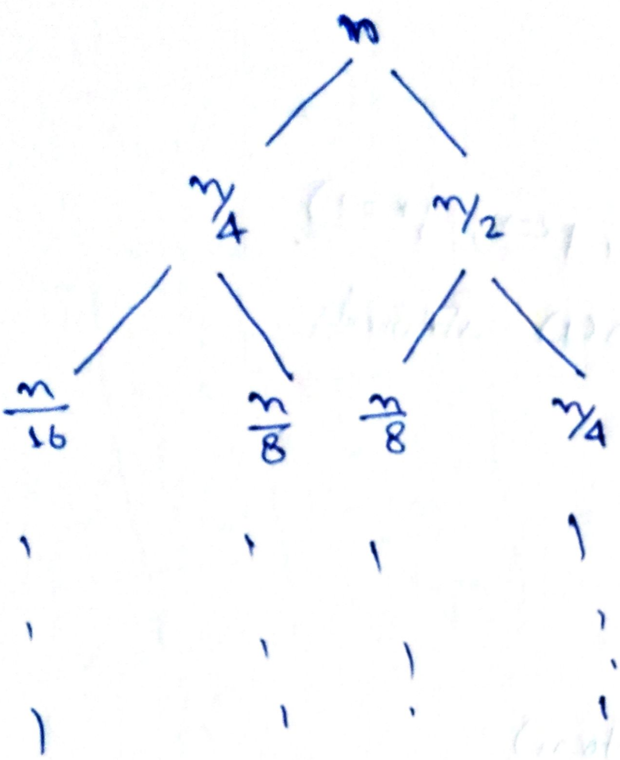
```
for (i to n)
    for (j to n)
        for (k to n)
            O(1) statements
```

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

~~for (int i=0; i<n;~~

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

Q.14) $T(n) = T(n/4) + T(n/2) + cn^2$



$\rightarrow cn^2$

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

$\rightarrow \frac{cn^2}{256} + \frac{cn^2}{64} + \frac{cn^2}{64} + \frac{cn^2}{16}$

$= \frac{25\,cn^2}{256}$

so $T(n) = cn^2 + \frac{5n^2}{16} + \frac{25n^2}{256} + \cdots)$

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

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

$= cn^2\left(\frac{1}{1-\frac{5}{16}}\right) = cn^2 \times \frac{16}{11}$

so T.C $\Rightarrow$ $\Theta(n^2)$

Q.15 ⇒

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

i	j	times
1	$1 \to n$	$n-1$
2	$1 \to n$	$(n-1)/2$
3	$1 \to n$	$(n-1)/3$
⋮	⋮	⋮
n	$1 \to n$	$n-1/n$
		$n \log n$

$$\boxed{T.C. \Rightarrow O(n \log n)}$$

Q.16 ⇒

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

Sol ⇒ Series $= 2, 2^k, 2^{2k}, 2^{3k}, \ldots,$

let last term be 2^{x*k}

$n = 2^{kx}$

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

Q.16 ⇒ for (int i= 2; i<=n; i = pow (i,k))

{ O(1);

}

$i = 2, 2^k, 2^{k^2}, 2^{k^3} \ldots, 2^{k^x}$

$n = 2^{k^x}$

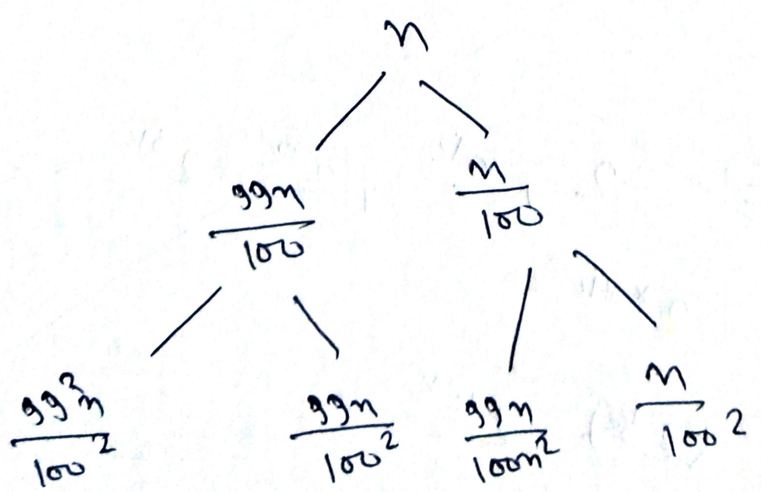
$\log n = k^x \log 2$

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

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

So T.C ⇒ $O(\log \log n)$

Q.17 ⇒ ~~$F(n) = T(n-1) + n$~~ $T(n) = T\left(\frac{99n}{100}\right) + T\left(\frac{n}{100}\right)$



If we take longer branch i.e. $\frac{99n}{100}$

T.C. $\Rightarrow \log_{\frac{100}{99}} n \simeq \log n$

$n = \left(\frac{99}{100}\right)^k \qquad k = \log_{\frac{100}{99}} n$

$T(n) = n \left(\log_{\frac{99}{100}}\right)^n = o(n \log_{99} n)$

Q18 → Increasing of growth.

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

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

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

$\log n! < 8^{2n} < n!$