Ans 1:

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

$i = 1, 2, 3, \ldots n$ until it becomes greater than or equal to n.

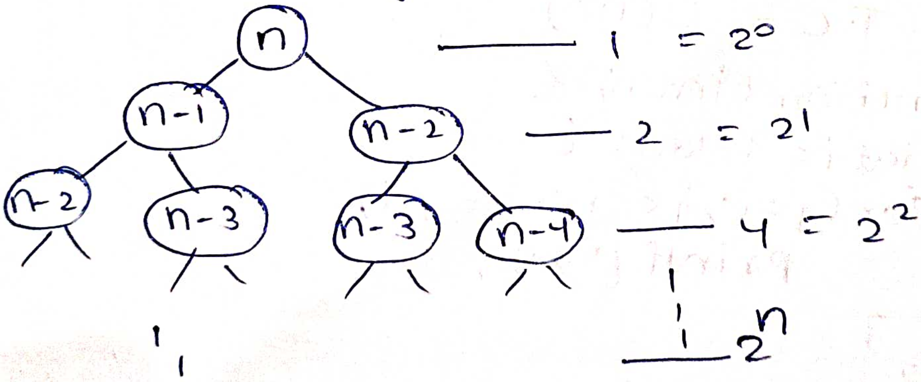
$\therefore \frac{x(x+1)}{2} \leq n$

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

Ans 2: Recurrence relation for recursive function that prints fibonacci series -

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

Using backward substitution it will cause problem so we will be using Tree's method.



$T = 1+2+4+ \dots + 2^n$

$a=1, \ r=2$

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

$\therefore O(2^{n+1}) = O(2 * 2^n) = O(2^n)$

Space complexity - As the maximum depth is proportional to N, hence space complexity is O(n).

3 (i) n log n

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

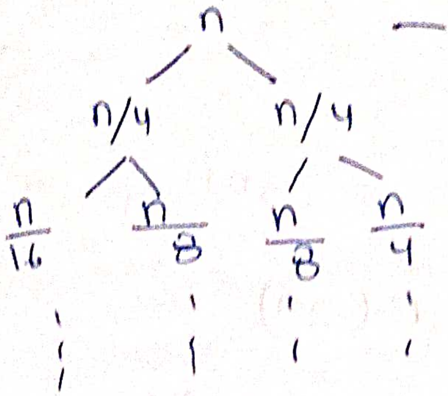
(ii) n^3

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

(iii) log (log n)

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

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



$— \; cn^2$

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

$— \; \frac{cn^2}{256} + \frac{cn^2}{64} + \frac{cn^2}{64} + \frac{cn^2}{16} = \frac{25}{256} cn^2$

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

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

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

$cn^2 \cdot \frac{16}{11} \quad \therefore \boxed{T.C = \Theta(n^2)}$

(5)

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

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

(6) $i = 2, 2^k, 2^{k^2}, 2^{k^3} \; \cdots \; 2^{k^n}$

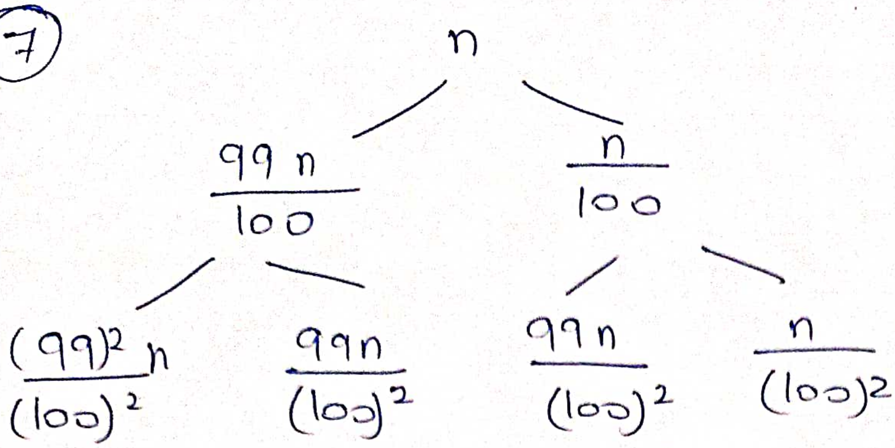
$n = 2^{k^x}$

$\log n = \cancel{x}^n \log_2$

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

$\therefore T.C = O(\log \log n)$

(7)



Taking longer branch $\frac{99n}{100}$

Time complexity $= \log_{\frac{100}{99}} n \simeq \log n$

$n = \left(\frac{99}{100}\right)^k$ or $k = \log\left(\frac{100n}{99}\right)$

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

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

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

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

Nandini