Time and space complexity

1.)

Ans

```
int sum=0
for (int i ; i<=n; i++) {
  for ( int J=1 ; J<=i ; J++) {
      sum++;
  }
}
```

The complexity would be $O(n^2)$ because there are 2 loops and in the worst case the loops will run $n \times n$ times.

This can be improved by using mathematical formulas.

2.) $T(n) = 3T(n-1) + 12n$; $T(0) = 5$

Ans

$T(0) = 3T(0-1) + 12 \times 0$

$= 3T(-1)$

$3T(-1) = 5$

$T(-1) = 5/3$

$T(1) = 3T(1-1) + 12(1)$

$= 3T(0) + 12$

$= 3 \times 5 + 12$

$T(1) \Rightarrow 15 + 12 \Rightarrow 27$

$T(2) = 3T(2-1) + 12 \times 2$

$\Rightarrow 3T(1) + 24$

$\Rightarrow 3 \times 27 + 24$

$81 + 24 \Rightarrow 105$

DATE / /20
PAGE NO.

3.) $T(n) = T(n-1) + c$ — i

Let the soln be - $O(n)$

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

$T(n-1) = T(n-2) + c$ — ii

putting (ii) in eqs (i)

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

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

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

similarly so on

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

$T(n) = T(1) + cn$

let $T(1)$ be constant

Than $T(n) <= cn$ Hence, $T(n) = O(n)$

4 $T(n) = 16T(n/4) + n^2 \log n$

Ans Master Theorem :

Here, $a = 16$; $b = 4$; $k = 2$

$a = b^k$

$16 = 4^2$ so, we need to compare p

$p = 1$ Than $\cancel{T(n) = \Theta(n^{\log_b a} \log\log n)}$

$\Rightarrow \cancel{n^{\log_4 16} \log \log}$

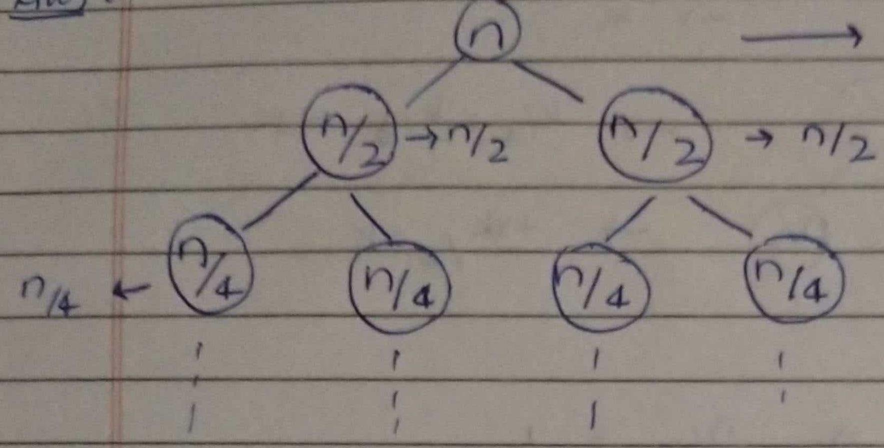
$T(n) = O(n^{\log_b a} \log^{p+1} n)$

$\Rightarrow O(n^{\log_4 16} \times \log^{1+1} n)$

$\Rightarrow O(n^2 \times \log^2 n)$

5c) $T(n) = 2T(n/2) + n$

<u>Ans:</u>



$\rightarrow n$

$\rightarrow n/2 + n/2 \Rightarrow n$

$\rightarrow n/4 \times 4 \Rightarrow n$

$$\frac{n}{2^0} + \frac{n}{2^1} + \frac{n}{2^2} + \frac{n}{2^4} \cdots\cdots \frac{n}{2^k}$$

at level k

$$\frac{n}{2^k} = 1 \Rightarrow n = 2^k \Rightarrow \log_2 n = k \log_2 2$$

$$\Rightarrow k = \log_2 n$$

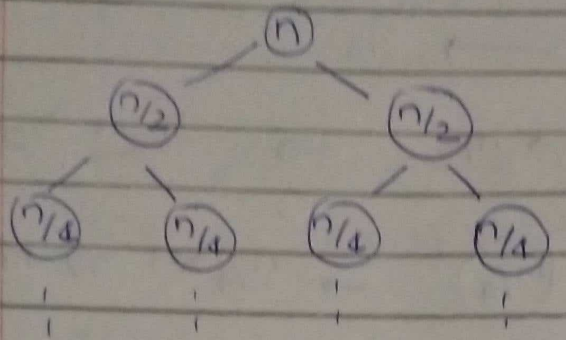
overall cost would be $k \cdot n$ ↱ cost at each level
↳ Levels

$$\Rightarrow \log_2 n \; n$$

$$\Rightarrow O(n \log_2 n)$$

6-) $T(n) = 2T(n/2) + k$

Ans



$\rightarrow k$

$\rightarrow k/2 + k/2 = 2k$

$\rightarrow 4 \times k/4 = k$

at level k' $\Rightarrow$ $n/2^k = 1$

$\Rightarrow n = 2^k$

$\boxed{\log_2 n = k}$

cost at each level $= k$

Total levels $= k$

$k(2^0 + 2^1 + 2^2 + \ldots\ldots \log_2 n)$

$\dfrac{k(2^{\log_2 n} - 1)}{1} \Rightarrow k(n^{\log_2 2} - 1)$

$\Rightarrow k(n-1)$

$\Rightarrow O(n)$