1) Substitution Method :-

It involves guessing the form of the solution and then using mathematical induction to find the constants and shows that the solution work.

1. Guess the solution
2. Use M.I & show that the guess is correct

Q. $T(n) = 2T\left(\frac{n}{2}\right) + n$

Sol → 1) Guess the solution.

$$\text{Let } T(n) = O(n\log n) \longrightarrow$$ Guess this using Master's Method

$$T(n) \leq c \cdot n\log n$$

2) Now we use M.I

(i) Base case :-

$n = 1$:- $T(1) \leq c \cdot 1\log 1$

$1 \leq 0$ false

$n = 2$:- $T(2) \leq c \cdot 2\log 2$

$2T\left(\frac{2}{2}\right) + 2 \leq c \cdot 2.$

$\Rightarrow 0 + 2 \leq c \cdot 2$ which is true.

$\therefore T(n) \leq c \cdot n\log n$ is true for $n = 2$.

(ii) Inductive step :- Now we assume it is true for $n/2$

i.e. $T\left(\frac{n}{2}\right) \leq c \cdot \frac{n}{2}\log\frac{n}{2}$ is true.

Now, we have show it is true for $n = n$.

$T(n) \leq cn\log n$

$T(n) \leq 2T\left[\frac{n}{2}\right] + n$

$\leq 2\left[c\left[\frac{n}{2}\right]\log\left(\frac{n}{2}\right)\right] + n$

$\leq cn\log\frac{n}{2} + n$

$\leq cn\log n - cn\log 2 + n$

$\leq cn\log n - cn + n$

$\leq cn\log n$ for $c \geq 1$

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

Scanned by TapScanner

Q $T(n) = T\left(\frac{n}{2}\right) + 1$

Sol → Let $T(n) = O(\log n)$. → Guess using Master Method

$T(n) \leq c \cdot \log n$.

Now we need to prove it correct by MI

Put this in the given recurrence eqn.

$$T(n) \leq c \log\left(\frac{n}{2}\right) + 1$$

$$\leq c \log \frac{n}{2} + 1$$

$$\leq c \log n - c \log 2 + 1$$

$$\leq c \log n - c + 1$$

$$\leq c \log n. \quad \text{for } c \geq 1.$$

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

Complexity of loops :-

1) Linear loop :-

```
for (int i=0; i<n; i++) {        (n+1)
        statements;              → n
}
```

```
for (int i=1; i<=n; i+=2) {
        Statements;              → n
}
```

$\boxed{\frac{n}{2}}$

2) Logarithmic loop :-
Multiply / divide

```
for (i = 1 ; i <= n ; i *= 2) {
    Statements;
}
```
→ $\log n$

iteration	value of i
Initial	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
|	
k^{th}	2^k

$\Rightarrow k^{th}$ iteration $= 2^k$

$n \geq 2^k$

$\log n \geq k$

```
for (i = n ; i >= 1 ; i /= 2) {
    Statements;
}
```
→ $\log n$

Scanned by TapScanner

3) Nested loop:-

(a) linear logarithmic:-

```
for (i=1; i<=n; i++) {        → n
    for(j=1; j<=n; j*=2) {    → logn
    }
}
```

$T(n) = n \log n$

(b) Quadratic :-

```
for (i=1; i<=n; i++) {     → n
    for(j=1; j<=n; j++) {  → n
    }
}
```

$T(n) = n * n = n^2$

(c) Dependent Quadratic:-

```
for (i=1; i<=n; i++) {      → n
    for(j=1; j<=i; j++) {   → n(n+1)/2
    }
}
```

⇒

iteration i	No. of steps	
Initial	1	
1	1	= 1
2	1,2	= 2
3	1, 2, 3	= 3
4	1, 2, 3, 4	= 4
⋮		
k	1, 2, 3, 4 - - k	= k.
⋮		
n	1, 2, 3, 4. - n	= n

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

$Avg = \frac{n(n+1)}{2 \times \cancel{n}} = \frac{n(n+1)}{2}$.

Scanned by TapScanner