Sol :-

AAA :-

1) Asymptotic notations.

2) Recurssion.

3) Recursive Relation.

4) 1) Master Method

2) Iterative Method

3) Substitution Method.

5) Graphs.

6) 1) BFS.

2) DFS.

7) strongly connected components in a Graph using Kosaraju's Algorithm.

8) Dynamic programming

9) Bellman ford

10) Greedy Approach

11) Prim's Algorithm

12) Minimum Spanning Tree.

14) Dijkstra for find single source shortest path.

13) Kruskal's Algorithm.

→ Asymptotic notations :-

n $\log n$ n^2 2^n

1) Big -oh. O

2) Big - omega. Ω

3) Theta. θ

1) Big -oh   → $f(n) = O(g(n))$.

$f(n) \le c * g(n) \quad \forall\ c,\ n \ge 0$

→ $f(n) = 2n+3$, O

→ $2n+3 \le \cancel{}\ 6n$, $n \ge 1$

$n \ge 1$

→ $\boxed{O(n)}$ Average Bound.

L.P

O

$$\boxed{1 < \log n < \sqrt{n} < n} < n\log n < n^2 < n^3 < \cdots\cdots n$$

Ω Upper Bound

→ $O(n\log n)$ | $O(n^2)$ $O(n^3)$.

$O(n)$

→ $O(\log n)$ $O(1)$ $O(\sqrt{n})$

Big-omega :-

Ω

$f(n) = \Omega(\log n)$

$\Rightarrow f(n) \geq c * g(n) \quad \forall c, n \geq 1$

$\Omega(\log n) \mid \Omega(\sqrt{n}) \mid \Omega(n)$

Theta :-

Bound theta

~~$c_1 \cdot g(n) \leq$~~

$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

LP

$\theta(n)$

UP

Recurssion

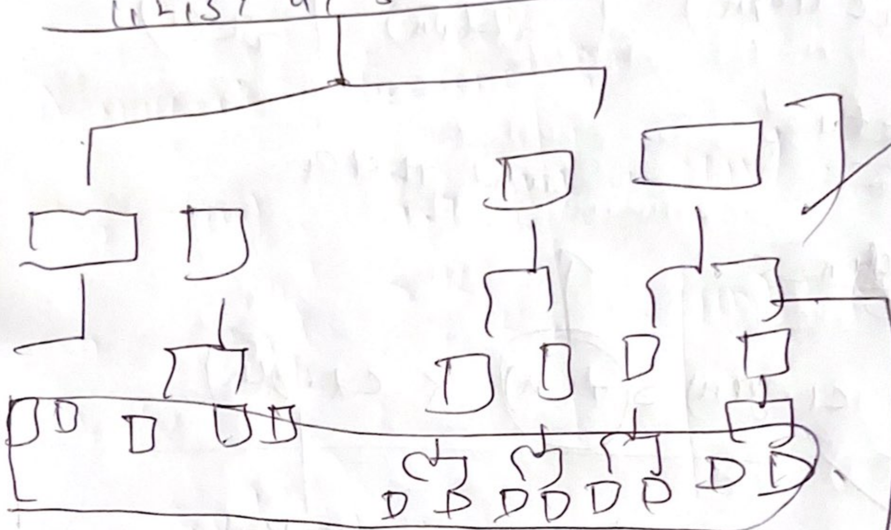
$f(n)$

C

1) mergesort

A $\underline{n}$ $\frac{n}{2}$

1, 2, 3, 4, 5, 6, 7



$$\boxed{T(n) = \underline{2} \cdot \underline{T}\left(\frac{n}{2}\right) + f(n)}$$

1) $n + n$

$T\left(\frac{n}{2}\right) \quad T\left(\frac{n}{2}\right) + T\left(\frac{n}{2^1}\right)$

$T\left(\frac{n}{4}\right) \quad T\left(\frac{n}{4}\right) \quad T\left(\frac{n}{4}\right) \quad T\left(\frac{n}{4}\right) + T\left(\frac{n}{2^2}\right)$

$T\left(\frac{n}{8}\right) \quad T\left(\frac{n}{8}\right) \quad T(n/8) \quad T(n/8) \quad T(n/8) \quad T(n/8) \quad T(n/8) \quad T(n/8) + T\left(\frac{n}{2^3}\right)$

$T(n/16) \quad T(n/16) \quad T(n/16) \quad T(n/16) \quad T(n/16) \quad T(n/16) \quad T(n/16) \quad T(n/16) + T(n/16)$

$T(n/16) \quad T(n/16) \quad T(n/16) \quad T(n/16) \quad T(n/16) \quad T(n/16) \quad T(n/16)$

$n \cdot K \quad n \quad T\left(\frac{n}{2^k}\right)$

$\rightarrow T\left(\frac{n}{2^k}\right) = T(1)$

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

$\frac{n}{n} = 1 \qquad \Rightarrow k = \log_2 n, \Rightarrow k = \log_2 n$

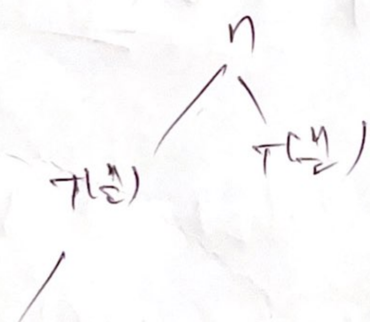
$\Rightarrow n \cdot k$

$\Rightarrow n \cdot \log_2 n$

$\Rightarrow O(n \log n)$

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

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



$\frac{n}{2^k} \qquad T(1)$

$k = 1 \cdot \log_2 n$

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

1) Master's Theorem
2) Iterative method.
3) Substitue Method.

1) Master's Theorem

$\Rightarrow T(n) = a.T(\frac{n}{b}) + f(n).$

$\Rightarrow \log_b^a, K, P \mid f(n) = \theta(n^K \log^P n)$

$\Rightarrow \log_b^a > K.$

Case-1 :- $T(n) = \theta(n^{\log_b^a})$

$\log_b^a = K.$

Case-2 :-

1) when $P \geq -1$

$\Rightarrow \theta(n^K \log^{P+1} n)$

2) when $P = -1$

$\Rightarrow \theta(n^K \log \log n)$

3) $p < -1$

$\theta(n^k)$

case-3 : $\log_b^a > k$

1) when $p \geq 0$

$\Rightarrow \theta(n^k \log^p n)$

2) when $p < 0$

$\Rightarrow \theta(n^k)$

$\log_b^a = 1$

$\log^0 n$

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

$\Rightarrow T(n) = aT(\frac{n}{b}) + f(n)$

$f(n) = \theta(n^k \log^p n)$

APP

$\Rightarrow a = 2, b = 2, k = 1, p = 0$

$\Rightarrow \log_b^a = 1 \mid k = 1$

$\Rightarrow \log_b^a = k$

$\Rightarrow \theta(n^1 . \log^1 n)$

2) $\Rightarrow$ $T(n) = 4T(\frac{n}{2}) + 1$

$\Rightarrow$ $f(n) = aT(\frac{n}{b}) + f(n)$

$f(n) = \theta(n^K \log^p n)$

$\Rightarrow$ $a=4$, $b=2$, $K=1$, $p=0$

$\Rightarrow$ $\log_b^a = \log_2^4 = \log_2^{2^2} \Rightarrow 2 \log_2^2$

$\Rightarrow$ 2 | $K=1$

$\log_b^a > K$

App/x - 1

$\theta(n^{\log_b^a})$

$\Rightarrow$ $\theta(n^2)$

3) $\Rightarrow$ $T(n) = 2T(\frac{n}{2}) + n^1 \log^1 n$

$a=2$ | $b=2$ $f(n) = \theta(n^K \log^p n)$

$\log_b^a = \log_2^2 = 1$ | $K=1$ $K=1$ | $p=0$

$\Rightarrow$ $f(n) \cdot \log n$

$\Rightarrow$ $\theta(n \log n \cdot \log n) = \theta(n \log^2 n)$

2) Iterative Method :-

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

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

$\Rightarrow T\left(\frac{n}{2}\right) = 2T\left(\frac{n/2}{2}\right) + \frac{n}{2}$

$\Rightarrow T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + \frac{n}{2}$

$\frac{n}{2^1}$

$T(n) = 2\left[2T\left(\frac{n}{4}\right) + \frac{n}{2}\right] + n$

$T(n) = 4T\left[\frac{n}{4}\right] + 2\cdot\frac{n}{2} + n$

$\Rightarrow T\left(\frac{n}{4}\right) = 2T\left[\frac{n/4}{2}\right) + \frac{n}{4}$

$T(n) = 4T\left[\frac{n}{4}\right] + n + n$

$\Rightarrow T\left(\frac{n}{4}\right) = 2T\left[\frac{n}{8}\right] + \frac{n}{4}$

$\frac{n}{2^2}$

$\Rightarrow T(n) = 4T\left[2T\left[\frac{n}{8}\right] + \frac{n}{4}\right] + n + n$

2^1

$2^2 T$

$\Rightarrow T(n) =$

$= 8T\left[\frac{n}{8}\right] + 4T\left(\frac{n}{4}\right) + n + n$

2^3

2^i

$T(n) = 8T\left(\frac{n}{8}\right) + n + n + n$

$\frac{n}{2^3}$

$\frac{n}{2^k}$

$$2^k\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} n$$

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

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

$T(n) = \boxed{T(\frac{n}{8})} + 1 + 1$

$T(n) = T(\frac{n}{8}) + 1 + 1 + 1$

$T(n) = T(\frac{n}{16}) + 1 + 1 + 1 + 1$

$T(n) = T(\frac{n}{2^k})$

$\Rightarrow T(\frac{n}{n}) = 1$

$\Rightarrow T(\frac{n}{2^k}) = T(1)$

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

$k = \log_2 n$

$T(\frac{n}{2}) = T(\frac{n}{4}) + 1$

$T(\frac{n}{4}) = T(\frac{n}{8}) + 1$

$T(\frac{n}{8}) = T(\frac{n}{16}) + 1$

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

~~Substitution~~ ~~Method~~

$T(n) = T(\sqrt{n}) + 1$

$T(n) = aT(\frac{n}{b}) + f(n)$

$\Rightarrow$ Let $n = 2^m$.

$\sqrt{n} = 2^{\frac{m}{2}}$

$\Rightarrow T(2^m) = T(2^{\frac{m}{2}}) + 1$

$\Rightarrow$ Let $T(2^m) = S(m)$,

$\Rightarrow T(2^{\frac{m}{2}}) = S(m/2)$,

$\Rightarrow S(m) = S(\frac{m}{2}) + 1$

$\Rightarrow T(n) = aT(\frac{n}{b}) + f(n)$

$\Rightarrow a = 1,\ b = 2 \quad k = 1,\ p = 0,$

$\Rightarrow \log_b^a = \log_2^1 = 0 \quad | \ k = 1$

$\log_b^a < k, \quad p \geq 0$

$\Rightarrow$ Applying master's theorem cases

$\Rightarrow \theta(n^k \log^p n) = \theta(n^1 \log^0 n)$.

$\Rightarrow \theta(n \log^0 n)$

Substitution Method :-

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

1) Guess it that $\Rightarrow O(n \log n)$.

2) we will prove that our guess is correct using mathmatecial induction

$m \to n$

$n \to n+1$

$1,2,3 \ldots n-1 \to 1,2 \ldots n$

$f(n) \leq n \log n$.

$\Rightarrow f(n) \leq c \cdot n \log n$.

$\forall \; m \leq n$

$\Rightarrow T(m) = c m \log m$. | $m \leq n$.

$m = \frac{n}{2} \leq n$

$\Rightarrow \boxed{T(\frac{n}{2}) \leq c\frac{n}{2} \log \frac{n}{2}}$

$\log \frac{a}{b} = \log a - \ldots$

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

$T(n) \leq cn \log \frac{n}{2} + n$

$T(n) \leq cn[\log n - \log_2 2] + n$.

$T(n) \leq cn \log n - \boxed{cn + n}$ $\Rightarrow T(n) \leq cn\log n$

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

1) Guess that solution is $\underline{O(n)}$.

2) we will prove this using induction.

$\Rightarrow T(n) = O(n)$

$\Rightarrow f(n) \leq c * g(n)$.

$\Rightarrow f(n) \leq c * n$

$\forall\ m \leq n \quad m = \frac{n}{2} < n$

$\Rightarrow f(\frac{n}{2}) \leq c * \frac{n}{2}$

$T(n) = \cancel{2} c \frac{n}{\cancel{2}} + n$

$f(n) \leq cn + n$

$\underline{f(n) \leq cn + n}$

Hence our Assumptions are wrong