Frac Knapsack Passing + algo:

Items →	1	2	3
(weight) w_i	18	15	10
P_i	25	24	15

Knapsack size = 20.

unit price:

1.39   1.6   1.5 ($\frac{15}{10}$)

solution: [বেশি যেটা সেটা নিব পুরা] $\boxed{2 > 3 > 1}$

item → 2 (1.6) → 1 which means 15.

~~item → 3 (1.5) → 1/2~~

so, (knapsack size − 15) = (20 − 15) = 5.

[এখনো নেওয়া যাবে]

item → 3 (1.5) → ½

$\left[5 - \left(\frac{1}{2} \times 10\right)\right] = 0.$

[আর নেওয়া যাবে না]

so, solution vector: $\left(0, 1, \frac{1}{2}\right)$.

profit: $\left[(25 * 0) + (24 * 1) + \left(15 * \frac{1}{2}\right)\right]$

= 31.5. (Ans).

algo wise; U = 20

w [15   10   18] (② ③ ①)

p [24   15   25]

so, 15 ≯ 20;

so, $x = \boxed{1} = 1$.

$U = 20 - 15 = 5.$

again,

10 > 5; yes

so, $x[2] = \frac{5}{10} = \frac{1}{2}$.

$U = 5 - 5 = 0$

again 18 > 0.

so, $x[3] = \frac{18}{0} = 0$.

Thus, $x\left[1, \frac{1}{2}, 0\right]$

rearrange to actual sequence; $x\left[0, 1, \frac{1}{2}\right]$

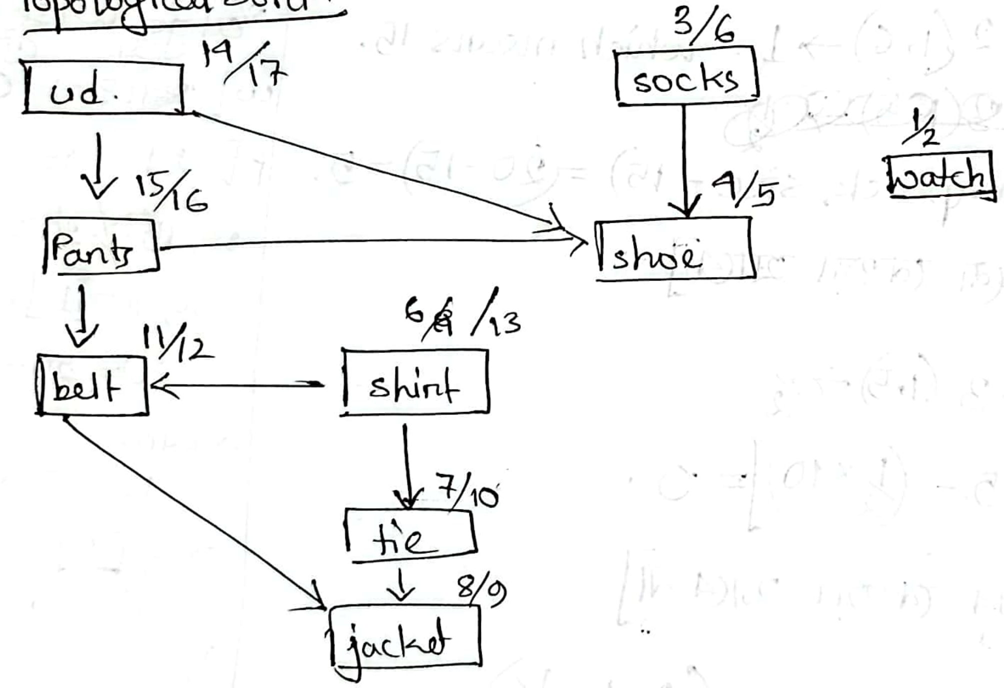
profit = 31.5. (same).

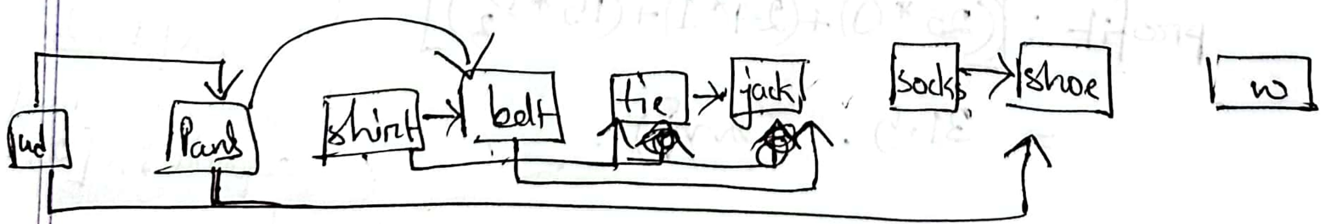
Frac knap algo:

```
Knapsack (m, n).        //p[i] and w[i] order is such that
  for i=1 to n          // p[i]/w[i] > p[i+1]/w[i+1] (बड़े to छोटे unit pr
      do x[i]=0 //initialize
  U=m.
  for i=1 to n
  {   if (w[i]>U) break;
        x[i]=1; U=U-w[i];
  }
  if (i<=n) then x[i]=U/w[i]
}.
```

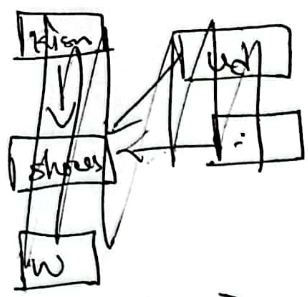
Topological Sort:

Sp-22
6/c





Sp-22 Q/e



Algo : (Topo)

O[V+E] 1. Call DFS(G) to compute finishing times f[v] for each vertex v.

O[V] 2. As each vertex ends, insert it to the front of a link list

O[1] 3. Return the list .

O[V+E]

DFS (Algo)

DFS(G)
for each vertex u ∈ V[G]
    do colour[u] ← white.
    π[u] ← NIL
time ← 0.

DFS(Algo)

0. DFS (source)
1. colour[source] = gray
2. For all adjacent edges from source to v
3.   v = adjacent edge node
4.   if colour[v] = white.
5.   DFS[v]
6.   end if
7. end for
8. colour[source] = black

O[V + E]

```
BFS(source)    O[V+E]
Q = queue()
level[ ] = -1 or ∞
Q.push[source).
Level[source] = 0
while Q not empty.
    u = Q.front().
    Q.pop().
    For all adjacent edges from u to v
        v = adjacent edge node.
        if level[v] = -1 or ∞
            Q.push(v)
            level[v] = level[u] + 1
        end if
    end for.
end while.
```

*** | Sp-22 7/a .

Subpaths of shortest paths are shortest paths:
Given a weighted, directed graph $G=(V,E)$ with weight f^n $w: E \rightarrow R$ let $p = \langle v_0, v_1, \dots, v_k \rangle$ be the shortest path from vertex v_0 to vertex v_k and for any i and j such that $0 \le i \le j \le k$, let $p_{ij} = \langle v_i, v_{i+1}, \dots, v_j \rangle$ be the subpath of p from vertex v_i to v_j. Then p_{ij} is a shortest path from v_i to v_j.

Decomposing path p into $v_0 \overset{p_{0i}}{\rightsquigarrow} v_i \overset{p_{ij}}{\rightsquigarrow} v_j \overset{p_{jk}}{\rightsquigarrow} v_k$ so that $w(p) = w(p_{0i}) + w(p_{ij}) + w(p_{jk})$. Now assum ~~path~~ p'_{ij} ~~from v_i t~~ is a path from v_i to v_j with weight

$w(P'_{ij}) < w(p_{ij})$. Then, $v_0 \overset{p_{0i}}{\rightsquigarrow} v_i \overset{p'_{ij}}{\rightsquigarrow} v_j \overset{v_{jk}}{\rightsquigarrow} v_k$ is a path from v_0 to v_k whose weight $w(p_{0i}) + w(P'_{ij}) + w(p_{jk})$ is less than $w(p)$, which contradicts the assuption that p is a shortest path from v_0 to v_k.

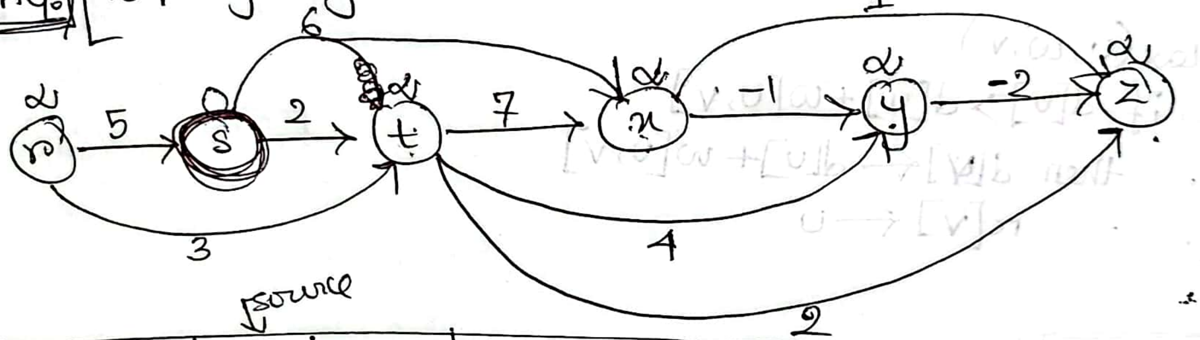
—— x ——

Single Source Shortest Paths — passings + Algo

input নেই : weighted + directed graph and one source vertex

output : shortest path weight, path ইত্যাদি.

DAG : [Topologically sorted থাকলে করি] + [neg value থাকলে problem নাই]



source

	r	s	t	x	y	z
initial	∞/N	0/N	∞/N	∞/N	∞/N	∞/N
r	∞/N	0/N	∞/N	∞/N	∞/N	∞/N
s	∞/N	0/N	2/s	6/s	∞/N	∞/N
t	∞/N	0/N	2/s	~~7/t~~ 6/s	6/t	4/t
x	∞/N	0/N	2/s	6/s	5/x	4/t
y	∞/N	0/N	2/s	6/s	5/x	3/y
z	∞/N	0/N	2/s	6/s	5/x	3/y

S paths:

r: (∞) → যাওয়া যায় না

s: x

s → t : 2

s → x : 6

s → x → y : 5

s → x → y → z : 3

→ DAG(G, w, s)

$O[V+E]$ 1. Topologically sort vertices of G

$O[V]$ 2. Initialize(G, s)

$O[V]$
3. for each vertex u, taken in toposorted order
4. do for vertex $v \in Adj[u]$
5. do Relax[u, v, w]

→ Initialize(G, s)

$O[V]$
1. for each vertex $v \in V[G]$
2. do $d[v] \leftarrow \infty$
3. $\pi[v] \leftarrow NIL$
4. $d[s] \leftarrow 0$

→ Relax(u, w, v)

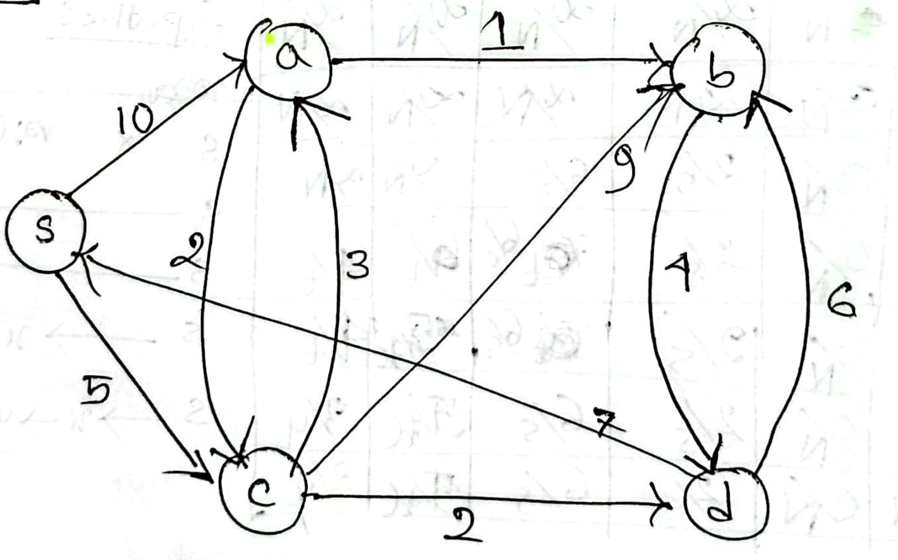
$O[1]$ 1. if $d[v] > d[u] + w[u,v]$

$O[1]$ 2. then $d[v] \leftarrow d[u] + w[u,v]$

$O[1]$ 3. $\pi[v] \leftarrow u$

$O[1]$

$\boxed{\text{Dijkstra}}$: [Cycle থাকলে করি - but- neg value থাকা যাবে না]



	s	a	b	c	d
initial	0/N	∞/N	∞/N	∞/N	∞/N
s	0/N	10/s	∞/N	5/s	∞/N
c	0/N	3+5=8/c	14/c	5/s	7/c
d	0/N	8/c	13/d	5/s	7/c
a	0/N	8/c	9/a	5/s	7/c
b	0/N	8/c	9/a	5/s	7/c

$s \to c \to a : 8$

$s \to c \to a \to b : 9$

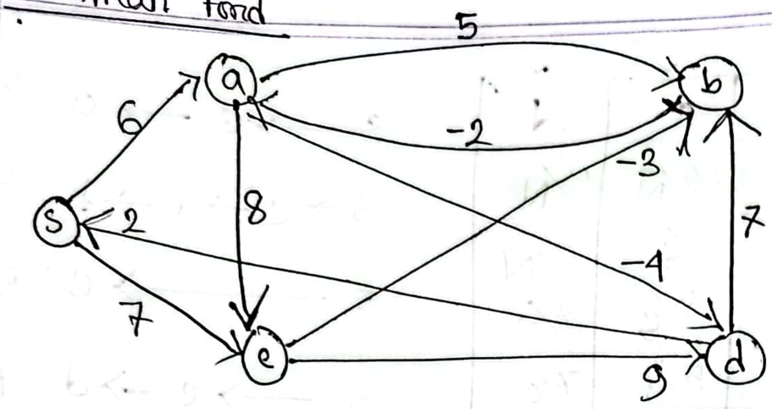
$s \to c : 5$

$s \to c \to d : 7$

Algo: Dijsktra(G, w, s)

1. Initialize(G, s)
2. $S \leftarrow \emptyset$
3. $Q \leftarrow V[G]$
4. while $Q \neq \emptyset$
5. do $u \leftarrow$ extract-min(Q)
6. $S \leftarrow S \cup \{u\}$
7. for each vertex $v \in Adj[u]$
8. do Relax(u, v, w).

Bellman Ford



(a, b)
(a, c)
(a, d).
(b, a).
~~(b, c)~~ (c, b)
(c, d)
~~(b, d)~~ (d, b)

$\boxed{(s,a) \\ (s,c)}$

	s	a	b	c	d
initialize	0/N	∞/N	∞/N	∞/N	∞/N
1st pass	0/N	6/s	∞/N	7/s	∞/N
2nd pass	0/N	6/s	11/a 4/c	7/s	2/a
3rd pass	0/N	2/b	4/c	7/s	2/a
4th pass	0/N	2/b	4/c	7/s	2/a

same 2[2]
off

$s \rightarrow c \rightarrow b \rightarrow a : 2$

$s \rightarrow c \rightarrow b : 4$

$s \rightarrow c : 7$

$s \rightarrow c \rightarrow b \rightarrow a \rightarrow d : 2$

Algo:

1. BellmanFord (G, w, s)

O[V] 2. Initialize (G, s)

O[V] 3. for i = 1 to |V[G]| - 1

O[VE] 4. do for each edge (u, v) ∈ E[G]

O[1] 5. do Relax (u, v, w)

O[VE] 6. for each edge (u, v) ∈ E[G]

O[1] 7. do if d[v] > d[u] + w[u, v]

O[1] 8. then return False

O[1] 9. return true.

O[VE]

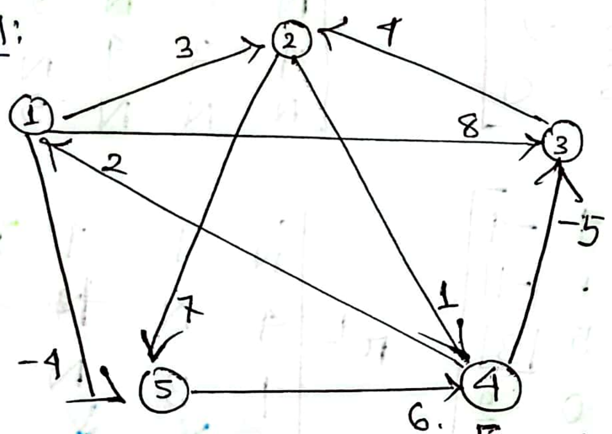
Floy warshall

Algo:

1. $n = W.rows$.
2. $D^{(0)} = W$.
3. for $k = 1$ to n
4. let $D^{(k)} = d_{ij}^{(k)}$ be a new $n \times n$ matrix
5. for $i = 1$ to n
6. for $j = 1$ to n
7. $d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$.
8. return $D^{(n)}$

Floyd warshall:

passing.



$$D^{(0)} = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad \pi^{(0)} = \begin{bmatrix} N & 1 & 1 & N & 1 \\ N & N & N & 2 & 2 \\ N & 3 & N & N & N \\ 4 & N & 4 & N & N \\ N & N & N & 5 & N \end{bmatrix}$$

$$D^{1} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \boxed{5} & -5 & 0 & \boxed{-2} \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array} \qquad \pi^{(1)} = \begin{bmatrix} N & 1 & 1 & N & 1 \\ N & N & N & 2 & 2 \\ N & 3 & N & N & N \\ 4 & 1 & 4 & N & 1 \\ N & N & N & 5 & N \end{bmatrix}$$

$$\pi_{ij}^{k} = \begin{cases} \pi_{ij}^{(k-1)} & \text{if } d_{ij}^{k-1} \leq d_{ik}^{k-1} + d_{kj}^{k-1} \\ \pi_{kj}^{k-1} & \text{if } d_{ij}^{k-1} > d_{ik}^{k-1} + d_{kj}^{k-1} \end{cases}$$

15 13 32 13 34 25

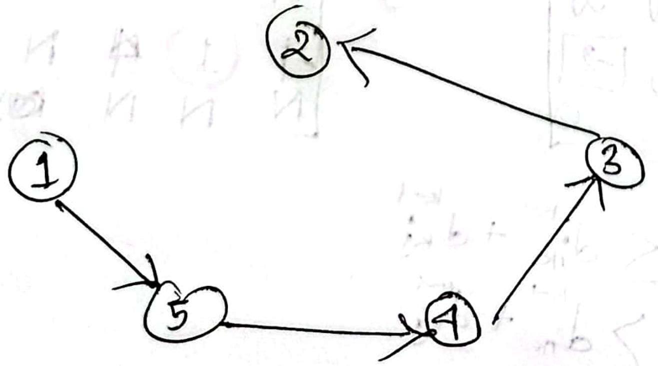
$$D^2 = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array} \qquad \pi^2 = \begin{bmatrix} N & 1 & 1 & 2 & 1 \\ N & N & N & 2 & 2 \\ N & 3 & N & 2 & 2 \\ 4 & 1 & 4 & N & 1 \\ N & N & N & 5 & N \end{bmatrix}$$

$$D^3 = \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & \boxed{-1} & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad \pi^3 = \begin{bmatrix} N & 1 & 1 & 2 & 1 \\ N & N & N & 2 & 2 \\ N & 3 & N & 2 & 2 \\ 4 & 3 & 4 & N & 1 \\ N & N & N & 5 & N \end{bmatrix}$$

$$D^4 = \begin{bmatrix} 0 & 3 & \boxed{-1} & 4 & -4 \\ \boxed{3} & 0 & \boxed{-4} & 1 & \boxed{-1} \\ \boxed{7} & 4 & 0 & 5 & \boxed{3} \\ 2 & -1 & -5 & 0 & -2 \\ \boxed{8} & \boxed{5} & \boxed{1} & 6 & 0 \end{bmatrix} \qquad \pi^4 = \begin{bmatrix} N & 1 & 4 & 2 & 1 \\ 4 & N & 4 & 2 & \boxed{\cancel{4}1} \\ 4 & 3 & N & 2 & \boxed{\cancel{4}1} \\ 4 & 3 & 4 & N & 1 \\ 4 & \boxed{3} & 4 & 5 & N \end{bmatrix}$$

$$D^5 = \begin{bmatrix} 0 & \boxed{1} & \boxed{-3} & \boxed{2} & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix} \qquad \pi^5 = \begin{bmatrix} N & 3 & 4 & 5 & 1 \\ 4 & N & 4 & 2 & 1 \\ 4 & 3 & N & 2 & 1 \\ 4 & 3 & 4 & N & 1 \\ 4 & 3 & 4 & 5 & N \end{bmatrix}$$

last সারি

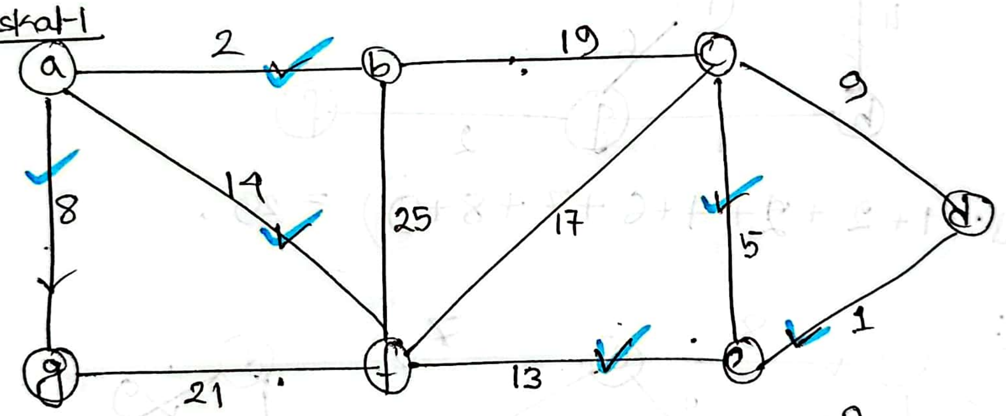


Kruskal's

Algo: kruskal()

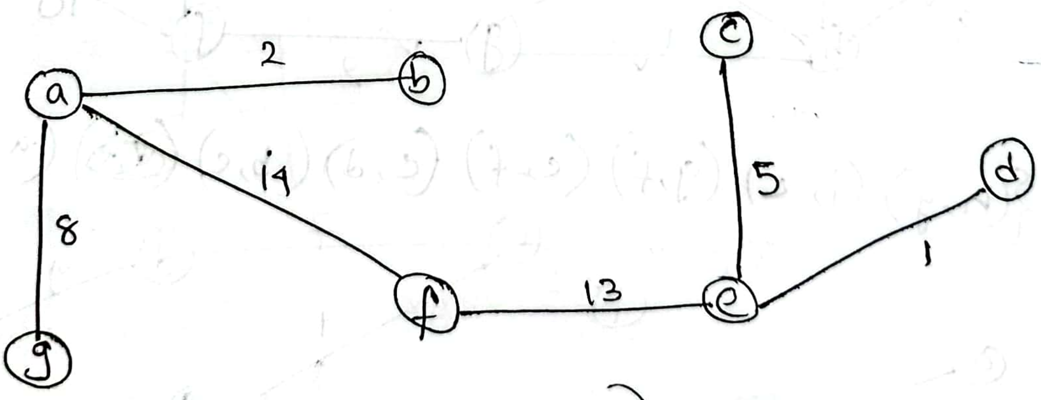
```
{  T = ∅
   for each v ∈ V
   Makeset(v);
   sort E by increasing edge w.
      for each (u,v) ∈ E
        if Findset(u) ≠ Findset(v)   // no cycle
        T = T ∪ {(u,v)};
        Union(Findset(u), Findset(v));
}
```

pass → kruskal-1



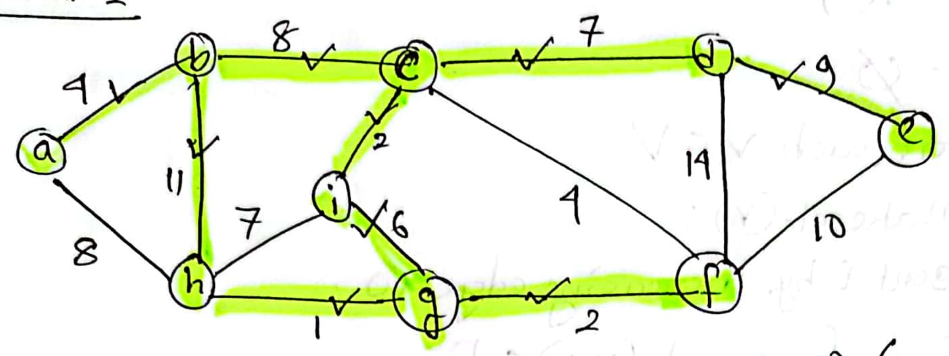
$T = \{(e,d)\ (a,b)\ (c,e)\ (a,g)\ (f,e)\ (a,f)\}$

MST :



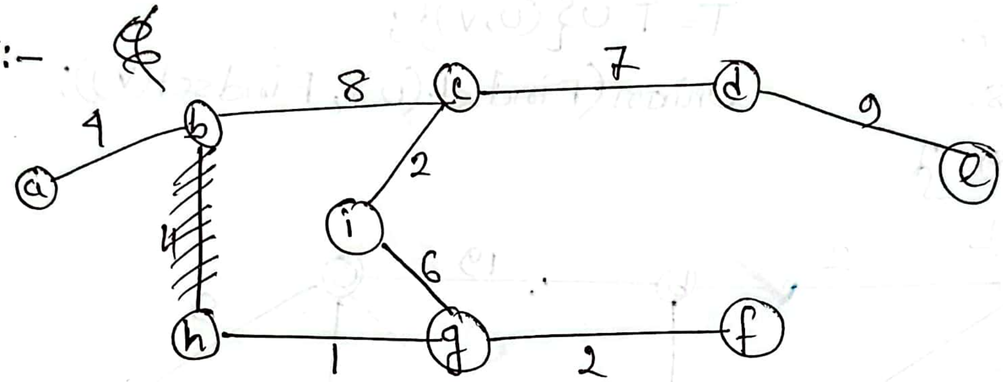
$\text{Total} = (1+2+5+8+13+14) = 43.$

Kruskal-2



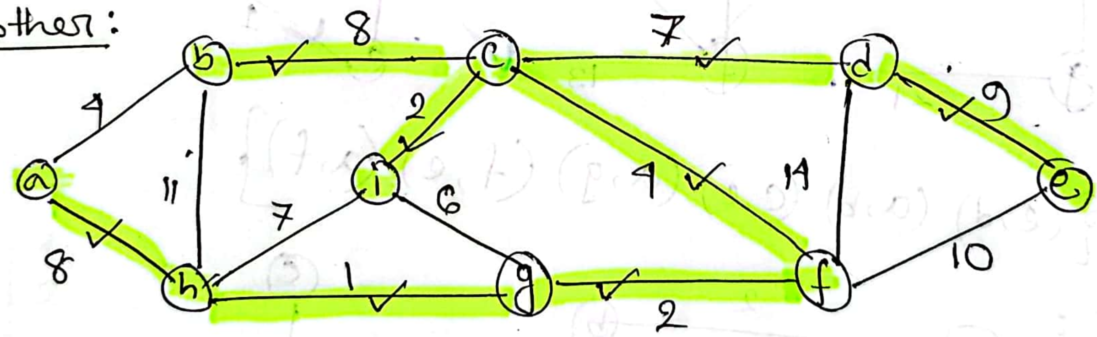
T= {(h,g) (i,c) (g,f) (a,b) (i,g) (c,d) (b,c) (d,e) (b,h)}

MST:-

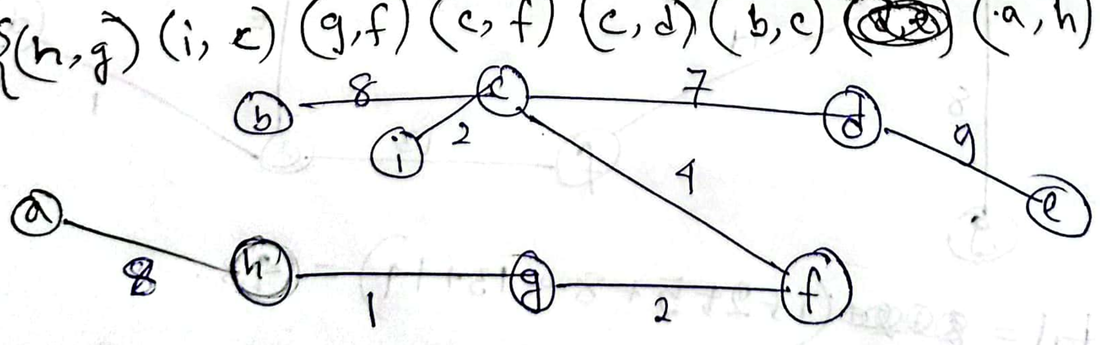


Total: (1+2+2+4+6+7+8+9) = 39.

another:

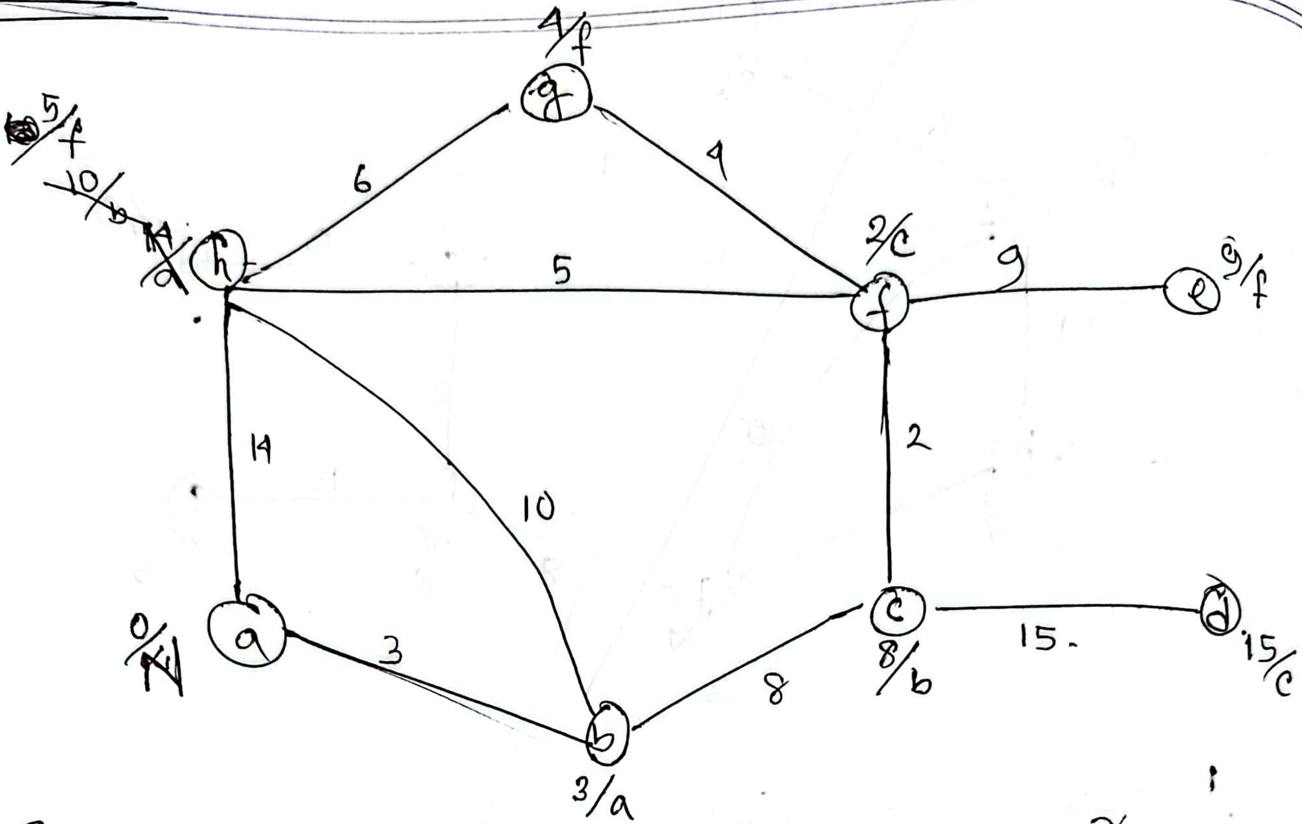


T= {(h,g) (i,c) (g,f) (c,f) (c,d) (b,c) (a,h) (d,e)}



Total = (1+2+2+4+7+8+8+9) = 41

Prims



T = { (a, b) (c, b) (c, d) (e, f) (c, f) (g, f) (h, f) }

Total cost : (3 + 8 + 15 + 9 + 2 + 4 + 5) = 46

MST ( )

for all node u
    weight[u] = ∞
    Q = all node.
    weight[root] = 0.
    P[root] = NULL.
    while (Q ≠ ∅)
        u = ExtractMin(Q)
        for each v ∈ adj[u]
            if (v ∈ Q and w(u,v) < weight[v]).
                p[v] = u.
                weight[v] = w(u,w).

MCM

input: $n=5$, $p = (\underset{P_0}{10}, \underset{P_1}{5}, \underset{P_2}{1}, \underset{P_3}{10}, \underset{P_4}{2}, \underset{P_5}{10})$

m

	1	2	3	4	5
1	0	50	150	50	190
2		0	50	30	90
3			0	20	40
4				0	200
5					0

S

	1	2	3	4	5
1		1	2	2	2
2			2	2	2
3				3	4
4					4
5					

$A_5 = (A_1 \; A_2)((A_3 \; A_4) \; A_5)$

formula:

$$m[i,j] = \begin{cases} 0 \\ \min\limits_{i \le k < j} \{ m[i,k] + m[k+1,j] + P_{i-1} P_k P_j \} \end{cases}$$

Let, $m(i,i) = 0$

$k=1$ $\quad m[1,2] = \min\{m[1,1] + m[2,2] + P_0 P_1 P_2\}$

$= \min\{0 + 0 + (10.5.1)\} = 50$

$k=1,2$ $\quad m[1,3] = \min\{m[1,1] + m[2,2] + P_0 P_1 P_3,$
$\quad m[1,2] + m[3,3] + P_0 P_2 P_3\}$

$= \min\{0 + 0 + (10.5.10), \; 50 + 0 + (10.1.10)\}$

$= \min\{500, 150\} = 150$.

$k=1,2,3$ $\quad m[1,4] = \min\{m[1,1] + m[2,4] + P_0 P_1 P_4,$
$\quad m[1,2] + m[3,4] + P_0 P_2 P_4,$
$\quad m[1,3] + m[4,4] + P_0 P_3 P_4\}$

$= \min\{0 + 30 + (10.5.2), \; 50 + 20 + (10.1.2);$
$\quad 150 + 0 + (10.10.2)\}$

$= \min\{130, 90, 350\}$

$= 90$

NOTE: [2,4] আগে করা লাগবে.
এভাবে solve.
না হলে মান miss হবে

$$K=2,\ m[2,3] = \min\{m[2,2] + m[3,3] + P_1 P_2 P_3\}$$
$$= \min\{0 + 0 + (5 \times 1 \times 10)\} = 50.$$

$$K=3,\ m[3,4] = \min\{m[3,3] + m[4,4] + P_2 P_3 P_4\}$$
$$= \min\{0 + 0 + (1 \cdot 10 \cdot 2)\} = 20$$

$$K=4,\ m[4,5] = \min\{m[4,4] + m[5,5] + P_3 P_4 P_5\}$$
$$= \min\{0 + 0 + (10 \cdot 2 \cdot 10)\} = 200$$

$$K=2,3\ \ m[2,4] = \min\{\min[2,2] + m[3,4] + P_1 P_2 P_4,$$
$$\min[2,3] + m[4,4] + P_1 P_3 P_4\}$$
$$= \min\{0 + 20 + (5 \cdot 1 \cdot 2),\ 50 + 0 + (5 \cdot 10 \cdot 2)\}$$
$$= \min\{30,\ 150\} = 30$$

$$K=3,4\ \ m[3,5] = \min\{m[3,3] + m[4,5] + P_2 P_3 P_5,$$
$$m[3,4] + m[5,5] + P_2 P_4 P_5\}$$
$$= \min\{0 + 200 + (1 \cdot 10 \cdot 10),\ 20 + 0 + (1 \cdot 2 \cdot 10)\}$$
$$= \min\{300,\ 40\} = 40$$

$$K=2,3,4\ \ m[2,5] = \min\{m[2,2] + m[3,5] + P_1 P_2 P_5$$
$$m[2,3] + m[4,5] + P_1 P_3 P_5$$
$$m[2,4] + m[5,5] + P_1 P_4 P_5\}.$$

$$= \min\{0 + 40 + (5 \cdot 1 \cdot 10).$$
$$50 + 200 + (5 \cdot 10 \cdot 10)$$
$$30 + 0 + (5 \cdot 2 \cdot 10)\}$$

$$= \min\{90,\ 750 + 130\} = 90.$$

$$K = 1,2,3,4 \quad m[1,5] = \min \begin{cases} m[1,1] + m[2,5] + P_0 P_1 P_5 \\ m[1,2] + m[3,5] + P_0 P_2 P_5 \\ m[1,3] + m[4,5] + P_0 P_3 P_5 \\ m[1,4] + m[5,5] + P_0 P_4 P_5 \end{cases}$$

$$= \min \begin{cases} 0 + 90 + (10 \cdot 5 \cdot 10), \\ 50 + 40 + (10 \cdot 1 \cdot 10) \\ 150 + 200 + (10 \cdot 10 \cdot 10) \\ 90 + 0 + (10 \cdot 2 \cdot 10) \end{cases}$$

$$= \min\{590, 190, 1350, 290\} = 190.$$

Optimal solution:

$$A_{15} = (A_1 \ A_2)((A_3 \ A_4) \ A_5)$$

mem (P)

```
n ← lenghth[P] − 1
for i=1 to n
    m[i,i] ← 0
for l=2 t n
    for i=1 to n−l+1
        j=i+l−1
        m[i,j] = ∞
        for k=i to j−1
            q=m[i,k] + m[k+1, j] + P_{i−1} P_k P_j
            if q<m[i,j]
                m[i,j]=q
                s[i,j]= K
```

LCS

subsequence ≠ substring

but substring = subsequence.

$X=\{A\ B\ C\ B\ D\ A\ B\}$ $\qquad$ $Y=\{B\ D\ C\ A\ B\ A\}$

Here, BBDA is not substring and subsequence of X but only subsequence for Y.

*** Optimal substructure of LCS.

Let, $X = \langle x_1, x_2 \ldots x_m \rangle$

$Y = \langle y_1, y_2 \ldots y_m \rangle$ be sequences.

and let $Z = \langle z_1, z_2 \ldots z_k \rangle$ be any LCS of X and Y.

1. $x_m = y_n$ then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}
2. $x_m \neq y_n$ then $z_k \neq x_m$ implies that Z is an LCS of X_{m-1} and Y.
3. $x_m \neq y_n$ then $z_k \neq y_n$ implies that Z is an LCS of X and Y_{n-1}

why?: ① If $z_k \neq x_m$, then we could append $x_m = y_n$ to Z to obtain a cs of X and Y of length k+1, contradicting the supposition that Z is a LCS of X and Y. Thus we must have $z_k = x_m = y_n$. Now, the prefix Z_{k-1} is a length (k-1) LCS of ~~X~~ X_{m-1} and Y_{n-1}. We wish to show it is a LCS. Suppose for the purpose of contradiction that there exists a cs W of X_{m-1} and Y_{n-1} with length greater than k-1, then appending $x_m = y_n$ to W produces a cs of X and Y whose length is greater than k, which is a contradiction.

② If $z_k \neq x_m$ then Z is a cs of X_{m-1} and Y; if there were a cs W of X_{m-1} and Y with length greater than k, then W would also be a cs of X_m and Y, contradicting the assumption that Z is an LCS of X and Y passing:

Y=ABCB.

X=BDCAB

	y_j	B	D	C	A	B
x_i	0	0	0	0	0	0
A	0	0	0	0	1 ↖	1 ←
B	0	1 ↖	1 ←	1 ←	1 ←	2 ↖
C	0	1 ↑	1 ↑	2 ↖	2 ←	2 ←
B	0	1 ↖	1 ←	2 ↑	2 ↑	3 ↖

length: 3.

reversed: BCB.

straight: BCB.

```
LCS (x, Y)
m = length(x)
n = length(y)
for i=1 to m
    c[i, 0] = 0
for j=1 to n
    c[0, j] = 0

for i=1 to m
    for j=1 to n
        if (x_i == y_j)
            c[i, j] = c[i-1, j-1] + 1
            b[i, j] = "↖"
        else if c[i-1, j] >= c[i, j-1]
            c[i, j] = c[i-1, j]
            b[i, j] = "↑"
        else if c[i, j] = c[i, j-1]
            b[i, j] = "←"
```

0,1 knapsack.

input = $n=4$. weight $=5$.
element (2, 3) (3, 4) (4, 5) (5, 6) $\boxed{(w, \text{profit})}$.

i \ w	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	$\boxed{3}$	3	3	3
2	0	0	3	4	4	$\boxed{7}$
3	0	0	3	4	5	7
4	0	0	3	4	5	7

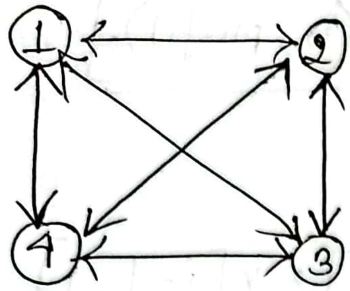
$v[i,j] \neq v[i-1,j]$

Ans: {1, 2}

```
01knapsack(n, w, E).
{ for w=0 to W
      v[0, w]=0
  for i=1 to n
      v[i,0]=0
  for i=1 to n
      for w=1 to W
          if wi<=w
             if bi + v[i-1, w-wi] > v[i-1, w]
                   v[i, w] = bi + v[i-1, w-wi]
          else
             v[i,w] = v[i-1, w]
}
```

~~else v[i,w] = [i-1,w];~~

TSP (DP)



	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

$g(2, \phi) = C_{21} = 5$
$g(3, \phi) = C_{31} = 6$
$g(4, \phi) = C_{41} = 8$.

$g(2, \{3\}) = C_{23} = 9$
$g(2, \{4\}) = C_{24} = 10$
$g(3, \{2\}) = C_{32} = 13$
$g(3, \{4\}) = C_{34} = 12$
$g(4, \{2\}) = C_{42} = 8$
$g(4, \{3\}) = C_{43} = 9$

$g(2, \{3\}) = C_{23} + g(3, \phi) = 9 + 6 = 15$
$g(2, \{4\}) = C_{24} + g(4, \phi) = 8 + 10 = 18$
$g(3, \{2\}) = C_{32} + g(2, \phi) = 13 + 5 = 18$
$g(3, \{4\}) = C_{24} + g(4, \phi) = 12 + 8 = 20$
$g(4, \{2\}) = C_{42} + g(2, \phi) = 8 + 5 = 13$
$g(4, \{3\}) = C_{43} + g(3, \phi) = 9 + 6 = 15$

$g(2, \{3, 4\}) = \min\{C_{23} + g(3, \{4\}), C_{24} + g(4, \{3\})\}$
$= \min\{9 + 20, 10 + 15\} = 25$.

$g(3, \{2, 4\}) = \min\{C_{32} + g(2, \{4\}), C_{34} + g(4, \{2\})\}$
$= \min\{13 + 18, 12 + 13\}$
$= \min\{31, 25\} = 25$

$g(4, \{2, 3\}) = \min\{C_{42} + g(2, \{3\}), C_{43} + g(3, \{2\})\}$
$= \min\{8 + 15, 9 + 18\}$
$= \min\{23, 27\} = 23$

$$g(1,\{2,3,4\}) = \min\{C_{12} + g(2,\{3,4\}),$$
$$C_{13} + g(3,\{2,4\}),$$
$$C_{14} + g(4,\{2,3\})\}$$
$$= \min\{10 + 25,\ 15 + 25,\ 20 + 23\}$$
$$= \min\{35, 40, 43\} = 35.$$

length : 35.

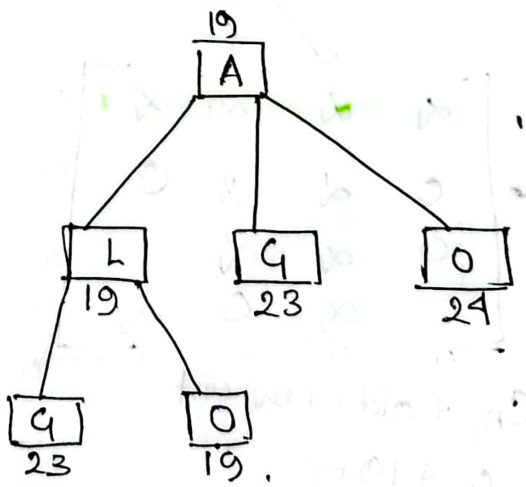
tour : $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$ (A)

Branch and Bound

$$cm = \begin{bmatrix} \alpha & 6 & 11 & 16 \\ 1 & \alpha & 5 & 6 \\ 2 & 9 & \alpha & 8 \\ 4 & 4 & 5 & \alpha \end{bmatrix}$$

	A	L	G	O
A	0	6	11	16
L	1	0	5	6
G	2	9	0	8
O	4	4	5	0

Tree :



row ~~column~~ reduce :

$$cm = \begin{array}{c|c} \begin{bmatrix} \alpha & 0 & 5 & 10 \\ 0 & \alpha & 4 & 5 \\ 0 & 7 & \alpha & 6 \\ 0 & 0 & 1 & \alpha \end{bmatrix} & \begin{matrix} 6 \\ 1 \\ 2 \\ 4 \end{matrix} \end{array}$$

(+)

13.

colourn reduce :

$$cm = \begin{bmatrix} \infty & 0 & 4 & 5 \\ 0 & \infty & 3 & 0 \\ 0 & 7 & \infty & 1 \\ 0 & 0 & 0 & \infty \end{bmatrix}$$

$$0 \quad 0 \quad 1 \quad 5 = 6.$$

So, row + colourn minimum cost $= 6+13 = 19$.

starting, reduced node:

(formula $= C_{ij}$ + oldcost + new cost) ***

$$A = \begin{bmatrix} \infty & 0 & 4 & 5 \\ 0 & \infty & 3 & 0 \\ 0 & 7 & \infty & 1 \\ 0 & 0 & 0 & \infty \end{bmatrix}$$

$$\begin{array}{c|c} \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 3 & 0 \\ 0 & \infty & \infty & 1 \\ 0 & \infty & 0 & \infty \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix} \\ \hline 0 \quad 0 \quad 0 \quad 0 & 0 \end{array} \qquad \begin{array}{c|c} \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 0 \\ 0 & 7 & \infty & 1 \\ 0 & 0 & \infty & \infty \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix} \\ \hline 0 \quad 0 \quad 0 \quad 0 & 0 \end{array} \qquad \begin{array}{c|c} \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 3 & \infty \\ 0 & 7 & \infty & \infty \\ 0 & 0 & 0 & \infty \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix} \\ \hline 0 \quad 0 \quad 0 \quad 0 & \end{array}$$

L (C_{AL} + old + new cost
$= 0 + 19 + 0$
$= 19$.

C(cost) $= 4 + 19 + 0$
$= 23$

O(cost) $= 5 + 19 + 0$
$= 24$

$$\begin{array}{cc} \begin{bmatrix} \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \cancel{\infty}\,0 \\ 0 & \infty & \infty & \infty \end{bmatrix} & \begin{matrix} - \\ - \\ 1 \\ 0 \end{matrix} \\ \begin{matrix} 0 & - & - & 0 \end{matrix} & 1 \end{array} \qquad \begin{array}{cc} \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ 0 & \infty & 0 & \infty \end{bmatrix} & \begin{matrix} 0\ - \\ - \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & - & 0 & - \end{matrix} & 0 \end{array}$$

$L \to G$ cost $= 3 + 19 + 1$

$= 23$

$L \to O$ cost $= 0 + 19 + 0 = 19$.

Path: $A \to L \to O \to G \to A$.

cost: $A \to L = 6$.

$L \to O = 6$

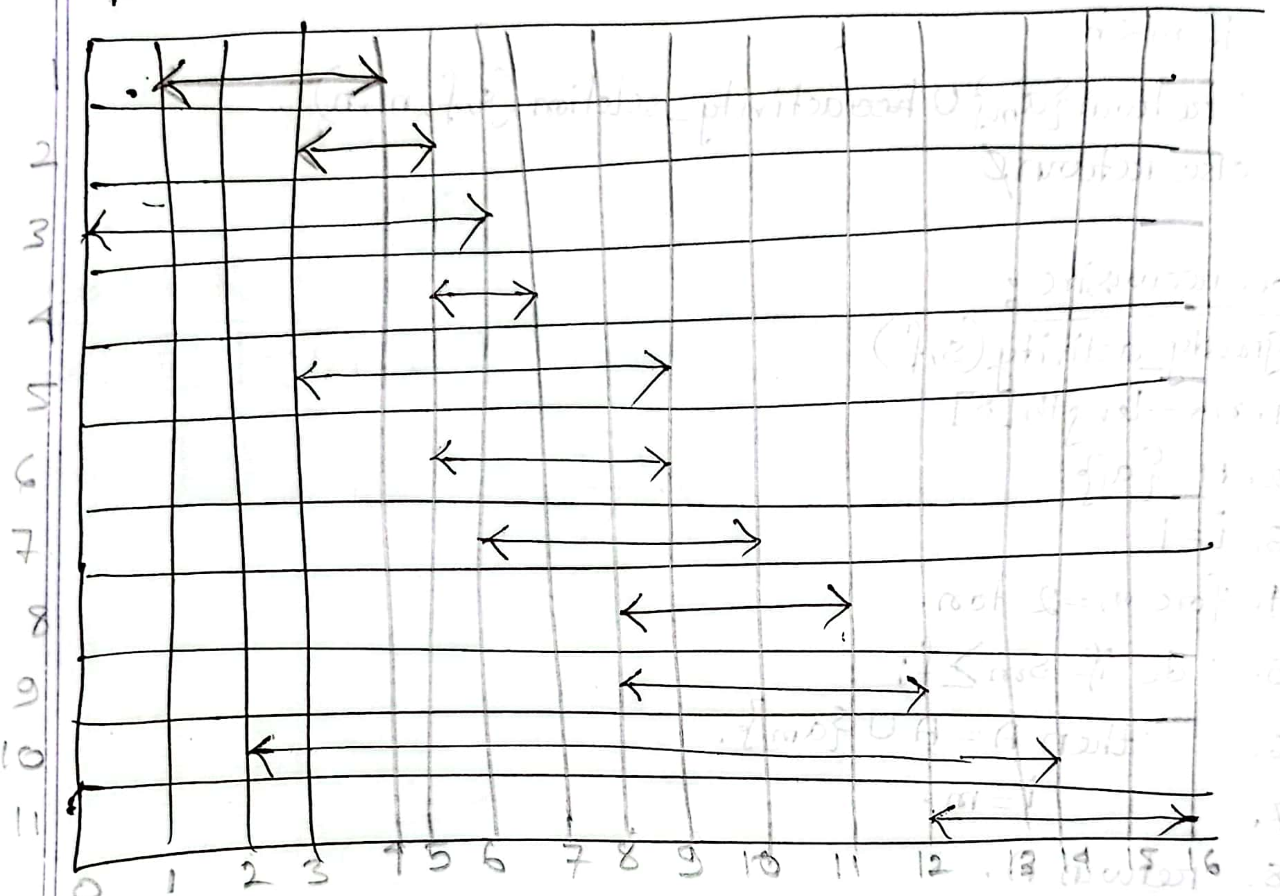
$O \to G = 5$

$G \to A = 2$.

total $= 19$.

Greedy activity selection (sp-22) 4/6.

	1	2	3	4	5	6	7	8	9	10	11
S	1	3	0	5	3	5	6	8	8	2	12
F	4	5	6	7	9	9	10	11	12	14	16



set1 = {1, 4, 8, 11}
set2 = {2, 4, 8, 11}
set3 = {3, 7, 11}

set 1 not unique
considering set 2 as Ans.

Why Greedy?

- It leaves as much opportunity as possible for the remaining activities to be scheduled.
- Greedy choice is the one that maximizes the amount of unscheduled time remaining.

recursive:

activity_selection(s, f, 0, n)

1. m = k+1
2. while m ≤ n and s[m] < f[k]
3. m = m+1
4. if m ≤ n
5. return {a_m} U ~~Rec~~ activity_selection (s, f, m, n)
6. else return Ø

non recursive:

greedy_activity(s, f)

1. n ← length[s]
2. A = {a_1}
3. i = 1
4. for m = 2 to n.
5. do if $s_m \geq f_i$
6. then A = A U {a_m}.
7. i = m.
8. return A.

Huffman Code :

fq	a(45)	b(13)	c(12)	d(16)	e(9)	f(5)
Fixed	000	001	010	011	100	101
Variable	0	101	100	111	1101	1100

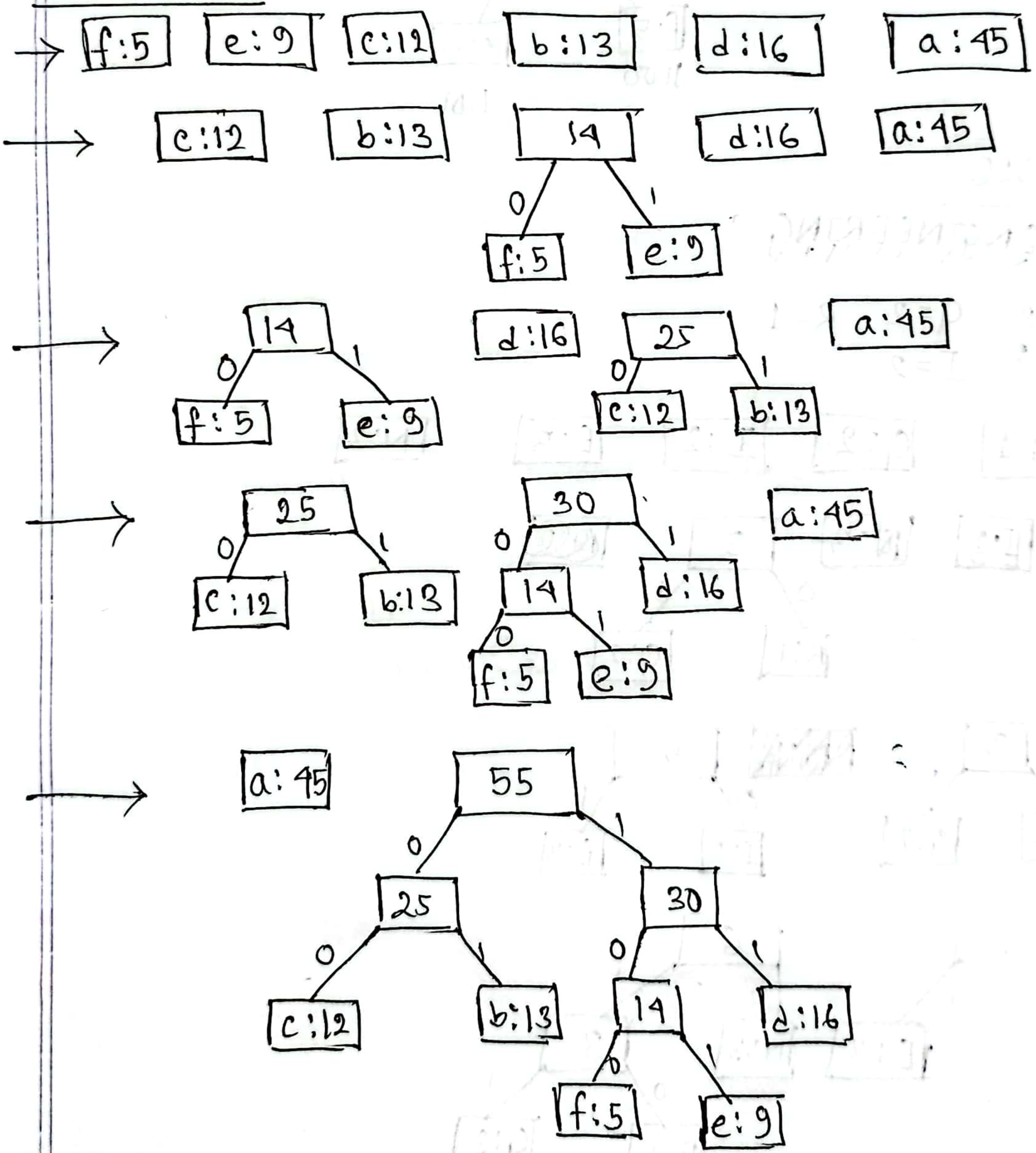
ASCII $\rightarrow 8 \times 1,00,000 = 8,00,000$ bits.

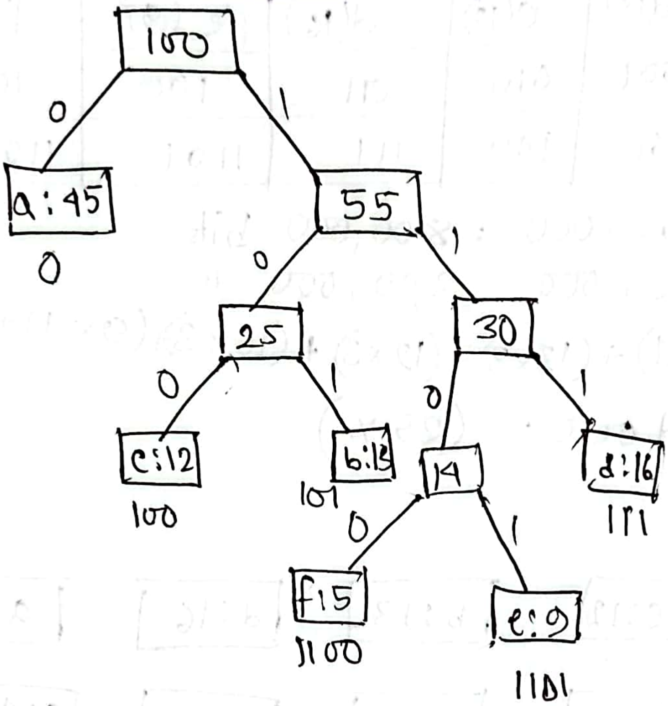
Fixed $\rightarrow 3 \times 1,00,000 = 3,00,000$ "

variable $\rightarrow (45 \times 1) + (13 \times 3) + (12 \times 3) + (16 \times 3) + (9 \times 4) + (5 \times 4) \times 1000$.

$= 224000$. (25 %)

Construction :

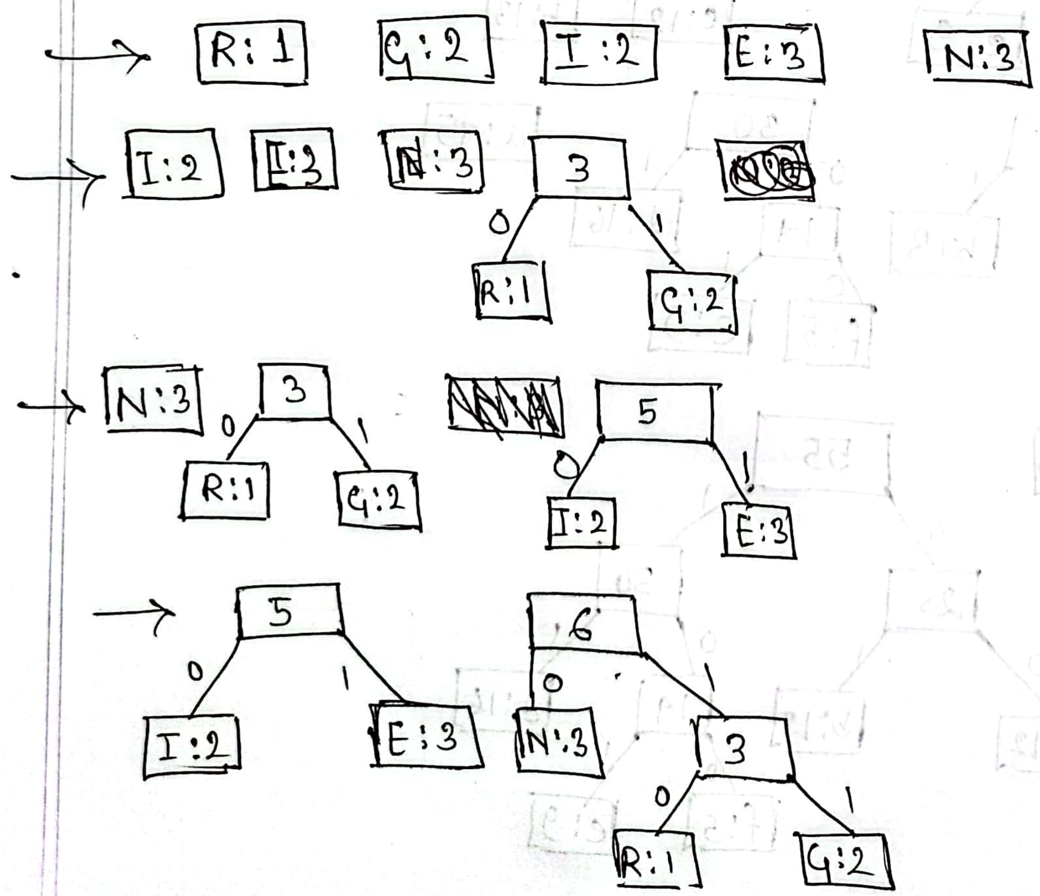


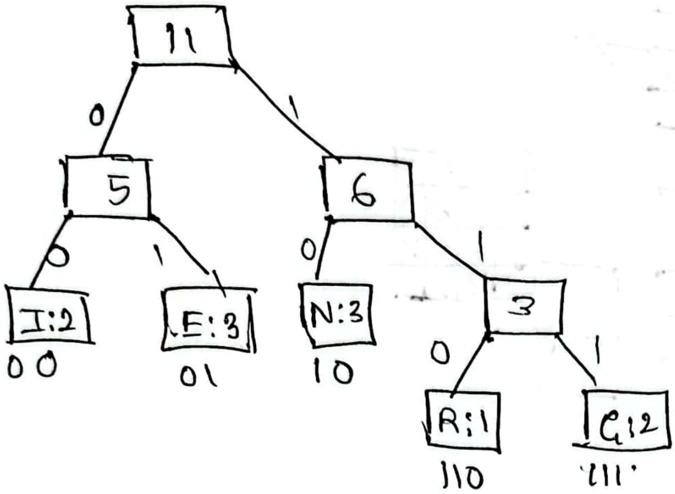


Fall-2020

3/a: ENGINEERING

E=3 G=2 R=1
N=3 I=2

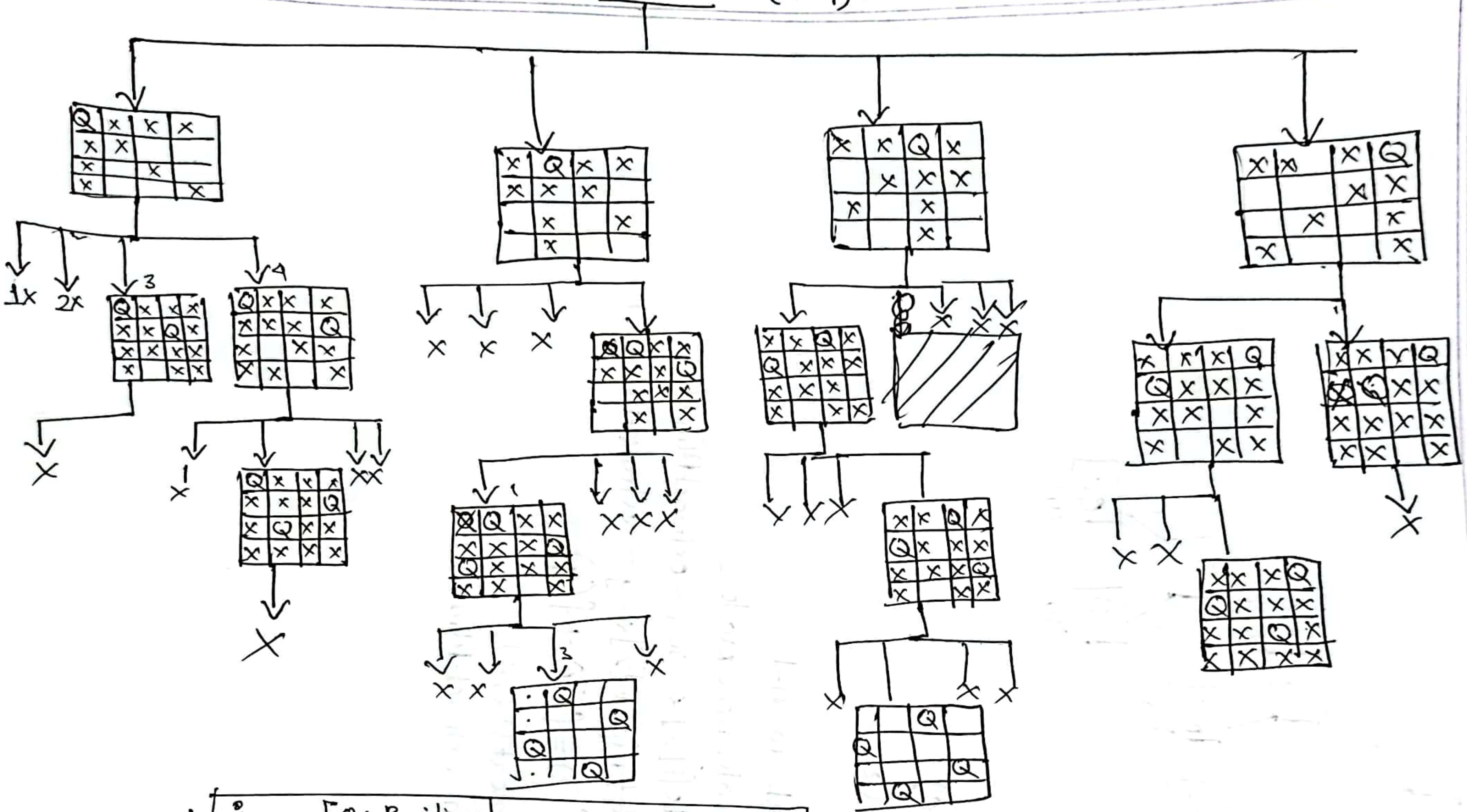




Code:

```
Huffman(C)
n = length(C)
Q = C
for i = 1 to n-1
    allocate new node z
        left[z] ← x ← extractMin(Q)
        right[z] ← y ← extractMin(Q)
        f(z) = f(x) + f(y)
    Insert(Q, z)
return extractMin(Q)
```

N-Queens (N=4).



solution:

[Q: Position]

1 (1:2) (2:4) (3:1) (4:3)

2 (1:3) (2:1) (3:4) (4:2)

```
QueenPlace(k,i)
{ for j=1 to k-1.
    if((x[j]=i) or (abs(x[j]-i)=abs(j-k)))
      then return false;
    return true,
}

NQueens(k,n)
{ for i=1 to n.
   { if QueenPlace (k,i) then
        { x[k]=i
          if (k=n) then x[1:n];
          else NQueens(k+1,n);
   } }
}.
```

Sub: Algorithm Vs Program:

Algorithm	Program
written while design time	implementation time
writer: needs domain knowledge	programmer
any language	programming language
HW + SW: independent	dependent
Can analyze	can test

*** why we need effective algo?

~~Reason 1~~:

we know insertion → $TC = c_1 n^2$

merge → $TC = c_2 n \log_2 n$

if $c_1 = 2$, insertion sort TC will be $2n^2$

if $c_2 = 50$, merge sort TC " " $50 n \log_2 n$

Now, consider comp. A and B.

A excutes 10^{10} instructions/sec

B " 10^7 " /sec

It seems like A is faster than B ~~for~~ (about 1000 times faster)

let $n = 10^7$ numbers (to be sorted)

run insertion in comp. A → $2 * (10^7)^2 / 10^{10}$ = 20,000 sec

=> 5.5 hours.

run merge " comp. B → $(50 * (10^7) \log_2 10^7) / 10^7$ = 1163 sec

=< 20 minutes

Hehe, Now, B runs ~~fas~~ 17 times faster than A

running time of an algo == f^n of input size n, $T(n)$

Common assumption → average case

Sub:

Day

Time : Date : / /

*** Why worst case better?
→ upper bound
→ Average case is as bad as worst.
→ frequently occur in database

ORDER OF GROWTH :

* leading term → ignore constant coeff + lower order terms
expl → $an^2+bn+c = \Theta(n^2)$

$n^2 = \Theta(n^2)$

* Efficiency → we got lower order of growth. [only true for large inputs]

→ TIME Fn :

$1 < \log n < n < n\log n < n^2 < n^3 < 2^n < n!$

const, log, linear, log linear, quadratic, cubic, expo, fact

ASYMPTOTIC NOTATION :

O [upper] worst $f(n) \leq c.g(n)$.
Ω [lower] be $f(n) \geq c.g(n)$
Θ [tight] ac $c_1 g(n) \leq f(n) \leq c_2 g(n)$

* $f(n) = 3n^2+4n+2$
$\therefore 3n^2+4n+2 \leq 3n^2+4n^2+2n^2$.
$\Rightarrow 3n^2+4n+2 \leq 9n^2$.

Sub: RECURRENCES.

Day

Time: Date: / /

Master method:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n) \quad \left[f(n) \cong \cancel{n\log}\ n^{\log_b a}\right]$$

Case: 1 $f(n) = O(n^{\log_b a - \varepsilon}) \rightsquigarrow T(n) = \Theta(n^{\log_b a})$

case: 2 $f(n) = \Theta(n^{\log_b a}) \rightsquigarrow T(n) = \Theta(n^{\log_b a} \log n)$

case: 3 $f(n) = \Omega(n^{\log_b a + \varepsilon}) \rightsquigarrow T(n) = \Theta(f(n))$.

Exples:

① $T(n) = 2T(n/2) + n$

$a = 2,\ b = 2,\ f(n) = n$

So, $\log_2 2 = 1 \rightsquigarrow n^{\log_2 2}$

So, $n^1 = f(n) = n$. [∵ case: 2]

$\therefore T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(n \log n)$.

② $T(n) = 2T(n/2) + n^2$

$a = 2,\ b = 2,\ f(n) = n^2$

So, $n^{\log_2 2} = n^1$ which is not equal to n^2

but $n^{\log_2 2 + 1} = n^2$ [∵ case: 3]

$\therefore T(n) = \Theta(n^2)$

③ $T(n) = 2T(n/2) + \sqrt{n}$

$a = 2,\ b = 2,\ f(n) = n^{\frac{1}{2}}$

So, $n^{\log_2 2} \Rightarrow n \neq n^{1/2}$

but $n^{\log_2 2 - 1} = n^{1/2}$ [∵ case: 1]

$\therefore T(n) = \Theta(n^{\log_2 2}) = \Theta(n)$

④ $T(n) = 3T(n/4) + n\lg n$

$a = 3,\ b = 4,\ f(n) = n\log n$

So, $\cancel{n\log n}\ n^{\log_3 4} = n^{0.793}$

but, $n^{0.793} \neq n\log n$.

$\cancel{\text{or}}, n^{0.793+\varepsilon} = n\log n$ [∵ case: 3]

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

⑤ $T(n) = 2T(n/2) + n\lg n$

$a = 2,\ b = 2,\ f(n) = n\lg n$

So, $n^{\log_2 2} \Rightarrow n \neq n\lg n$.

but, $n^{1+\varepsilon} = n\lg n$ [∵ case: 3]

$\therefore T(n) = \Theta(n\lg n)$.

we know $\Theta \to c_1 g(n) \leq f(n) \leq c_2 g(n)$

upper bound:

$15n^2+5n+5 \leq c_2 \cdot g(n)$.

$\Rightarrow 15n^2+5n+5 \leq 15n^2+5n^2+5n^2$.

$\Rightarrow 15n^2+5n+5 \leq 20n^2$

let, $c_2=20$ here. so, $O(n^2)$

Lower bound:

$15n^2+5n+5 \geq c_1 g(n)$

$15n^2+5n+5 \geq 15n^2$.

let, $c_1=15$. and $n_0=1$ for all $n \geq 1$. so, $\Omega(n^2)$

so, $\Theta(n^2)$.

Substitution:

$T(n) = 2T(\lfloor n/2 \rfloor) + n$

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

we have to prove, $T(n) \leq cn\log n$.

$T(\lfloor n/2 \rfloor) \leq c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor$

$T(n) \leq 2\{c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor\} + n$

$\leq cn\log(n/2) + n$

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

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

$\leq cn\log n$.

Sp-22

$T(\lfloor n/2 \rfloor) \leq c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor$

$T(n) \leq 4\{c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor\} + cn$

$\leq cn\log(n/2) + cn$

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

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

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