Implementatn

1. Read
2. test () // GS == IS
3. cost () // Misplaced tiles
4. findspace () // x, y
5. move (char) {
   temp[ ][ ] = 0

   up (x != 0)

   d (x != n-1)

   L (y != 0)

   R (y != n-1)

PAGE No.

DATE / /

Travelling Salesman Problem (TSP)

Branch & Bound

→ $G = (V, E)$ be a directed graph defining instance of TSP.

→ c_{ij} = cost of edge (i,j), $(i,j) \in E$
= ∞, $(i,j) \notin E$

→ Total no. of vertices $= |V| = n$.

→ without loss of generality, it can be assumed that every tour starts and ends at vertex 1.

→ So the solution space, S,

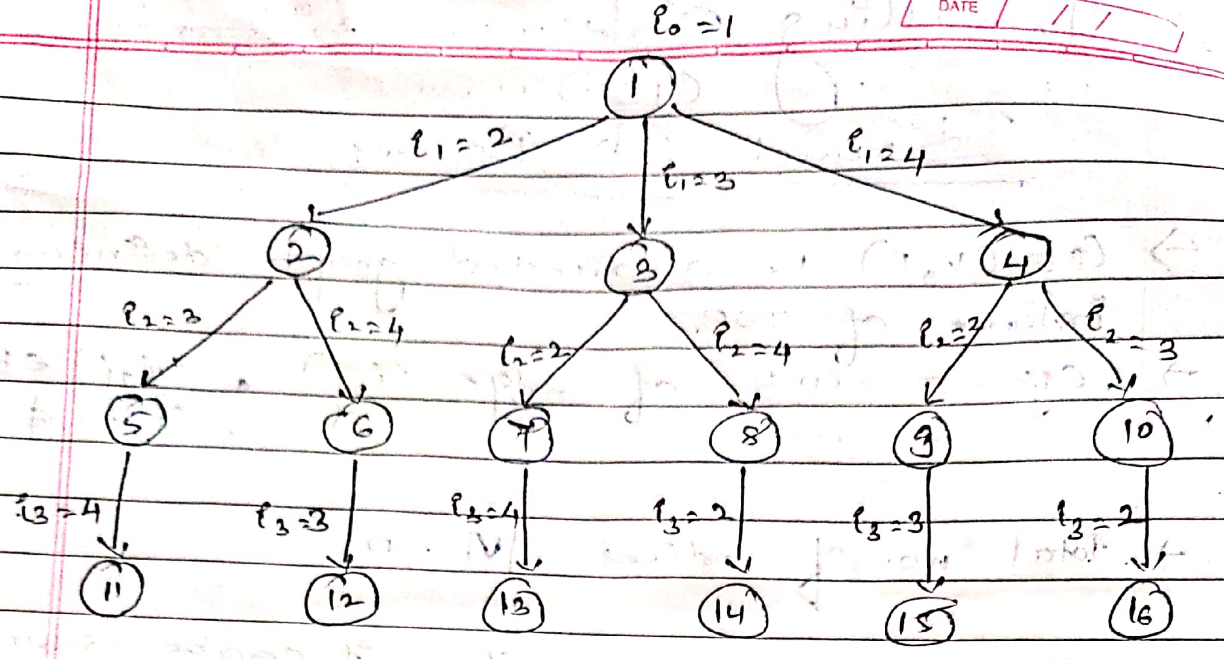
$S = \{1, \pi, 1 \mid \pi$ is a permutation of $(2, 3 \ldots n)\}$

$\therefore |S| = (n-1)!$

→ size of S i.e $|S|$ can be reduced by restricting 'S' so that $(1, i_1, i_2 \ldots i_{n-1}, 1) \in S$ iff,

$$(i_j, i_{j+1}) \in E \quad \text{and} \quad i_0, i_n = 1$$

→ Consider, a state space tree for TSP with a complete graph of $n = 4$ and $i_0 = i_4 = 1$



* Least Cost Branch & Bound

→ In LCBB, a cost function is defined such that, solution node with least cost correspond to a shortest tour in G.

→ cost(A) = length of tour defined by the path from root to node A (If A is leaf node)

= cost of minimum-cost leaf in the subtree A, (If A is not a leaf node).

→ Cost function alws to find a lower bound of TSP tour i.e an estimate of tour cost using Reduced cost matrix corresponding to G.

→ What is reduced cost matrix?

- A column (row) is said to be reduced iff it contains atleast one zero & rest non-negative entries.
- A matrix is reduced iff every row & column is reduced.

→ Why reduce cost matrix?

- Every tour includes exactly one edge $\langle i,j \rangle$ with $i=k$, $1 \le k \le n$ and exactly one edge $\langle i,j \rangle$ with $j=k$, $1 \le k \le n$
- Subtracting a constant 't' from every entry in one column or one row of cost matrix reduces the length of every tour by exactly t.
- A minimum cost tour remains minimum cost tour following this operation.

- If t is chosen to be minimum entry in row i (column j), then subtracting it from all entries in row i (column j) adds a zero in row i (column j).

- Repeat row & column reduction, till cost matrix is completely reduced.

→ Total row & column reduction is ~~the~~ a lower bound on length of a minimum cost tour.

- In state space tree, cost of root node i.e cost of node(1) is total reduction in adjacency matrix. say x.
- Hence, all tours in the original graph have a length atleast x.

Scanned by CamScanner

→ How to calculate cost of every node with reduced cost matrix in the travelling salesman state space tree ?

- Let M be the reduced cost matrix for node R.
- Let S be the child of R such that tree edge <R, S> corresponds to edge <i, j> in tour.

- If S is not a leaf node then reduced cost matrix is obtained by.

① Save M[i, j]

② Change all entries in row i and column j of M to ∞ (prevent use of edge leaving vertex i or entering vertex j)

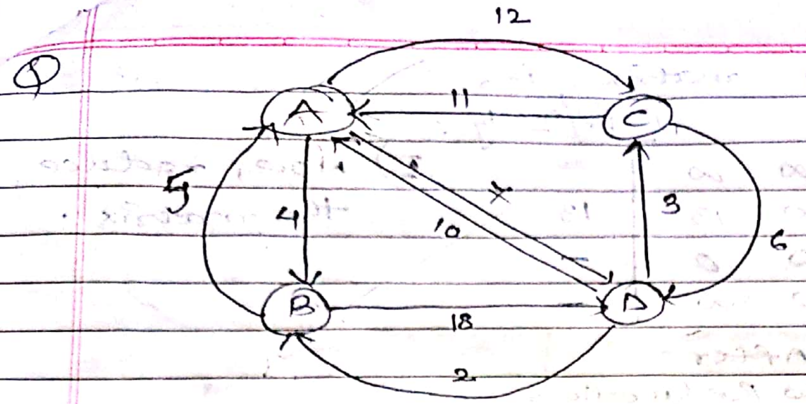
③ set M[j, 1] to ∞ (prevents use of edge <j, 1>)

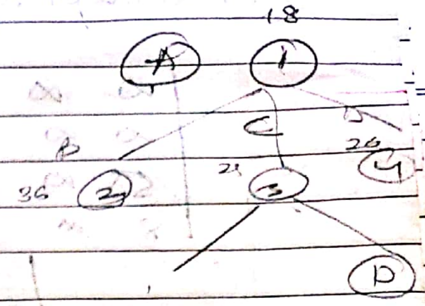
④ Now, reduce rows & columns in the resulting matrix except for rows and columns containing only ∞ or ~~Atleast~~ a zero.

→ Cost of Node S is given by,

cost (S) = cost (R) + reduction (step 4) + M[i, j] (step 1)

Q





Solution :- step 1 : Initial cost matrix & its Reduction.

				Reduce by
∞	4	12	7	4
5	∞	∞	18	5
11	∞	∞	6	6
10	2	3	∞	2

After row reduction

∞	0	8	3
0	∞	∞	13
5	∞	∞	0
8	0	1	∞
Reduce by –	–	1	–

After column reduction →

∞	0	7	3
0	∞	∞	13
5	∞	∞	0
8	0	0	∞

M

Cost (1) = Reduction = (4+5+6+2) + 1 = 18

Step 2 :- Select vertex B, Node 2 (A→B)

1) save, M [A, B] = 0
2) set row A & column B as ∞
3) Make M [B, A] = ∞

current Start vertex

PAGE No.
DATE / /

∴ Resulting matrix is,

$$\begin{array}{c|c} \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 13 \\ 5 & \infty & \infty & 0 \\ 8 & \infty & 0 & \infty \end{bmatrix} & \begin{array}{c} \text{Reduce by} \\ - \\ 13 \\ - \\ - \end{array} \end{array}$$

3) Now, reduce the matrix.

After Row Reduction

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ 5 & \infty & \infty & 0 \\ 8 & \infty & 0 & \infty \end{bmatrix} \xrightarrow{\text{After column reduction}} \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ 0 & \infty & \infty & 0 \\ 3 & \infty & 0 & \infty \end{bmatrix}$$

Reduce by: 5 — — —

$$cost(2) = cost(1) + Reduction + M[A,B]$$
$$= 18 + (13+5) + 0 = 36$$

Step 3 :- select vertex C : Node (3) , (A → C)

1) Save, M[A,C] = 7.
2) Set row A & column C as ~~set~~ ∞
3) Set M[C, A] = ∞

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 13 \\ \infty & \infty & \infty & 0 \\ 8 & 0 & \infty & \infty \end{bmatrix}$$

No row or column reduction required.

$$cost(3) = cost(1) + reduction + M[A,C]$$
$$= 18 + 0 + 7 = 25$$

Step 4 : Select vertex D, Node (4), (A → D)

(1) Save . M[A, D] = 3

(2) Make row A & column D as ∞

(3) Set M[D, A] = ∞

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ 5 & \infty & \infty & \infty \\ \infty & 0 & 0 & \infty \end{bmatrix} \quad \begin{array}{c} \text{Reduce by} \\ - \\ - \\ 5 \\ - \end{array}$$

After row Reduction

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ \infty & 0 & 0 & \infty \end{bmatrix}$$

- No column reduction required.

cost (4) = cost (1) + reduction + M[A, D]

= 18 + 5 + 3 = 36

∴ cost (3) i.e cost of node 3 is least cost, ∴ we will choose path A → C

Step 5 : select vertex B, Node (5), A → C → B

(1) Save, M[C, B] = ∞

(2) set row C & column B as ∞

(3) set M[B, A] = ∞

∴ Resulting matrix is,

Reduce by :

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 13 \\ \infty & \infty & \infty & \infty \\ 8 & 0 & \infty & \infty \end{bmatrix} \begin{matrix} \rightarrow \\ 13 \\ - \\ - \end{matrix}$$

↓ After row reduction

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty \\ 8 & \infty & 0 & \infty \end{bmatrix} \xrightarrow[\text{reduction}]{\text{After column}} \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty \\ 0 & \infty & 0 & \infty \end{bmatrix}$$

$8 \quad - \quad - \quad -$

cost(5) = cost(3) + Reductions + M[C, B]

= 25 + (13 + 8) + ∞

= ∞

step6 : select vertex D, Node (6), A → C → D

1. Save, M[C, D] = 0
2. set row C & column D as ∞
3. set M[D, A] = ∞

Resulting matrix is.

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \end{bmatrix}$$

No column & row reduction required.

cost(6) = cost(3) + Reductions + M[C, D]

= 25 + 0 + 0 = 25

Step 7: select vertex B, Node (7), A→C→D→B

① save, M[D,B] = 0
② set column B & row D as ∞
③ Set B→A to ∞.

Resulting matrix is,

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix}$$

No column & row ~~matrix~~ reduction required.

∴ cost (7) = cost (6) + reduction + M[D,B]
$= 25 + 0 + 0 = 25$

∴ Optimal path is
A→C→D→B→A with cost = 25