L-21

Date
Page

Solu^n of Shortest Path Problem:

Problem: For a connected edge weighted graph G and a source vertex $u \in V(G)$, find the minimum weight of a path from u to every other vertex in G. Let $\alpha : E(G) \to \mathbb{N}$ be the weight function.

Dijkstra's Algorithm:

(1) Set $S = \phi$ which will store the vertices of G for which a shortest path has been found.
(At the end of the process S will contain all vertices of G since G is connected.)

(2) Set $t(u) = 0$ and $t(v) = \infty$ for all $v \in V(G), v \neq u$ and add u to S.
→ leveling from

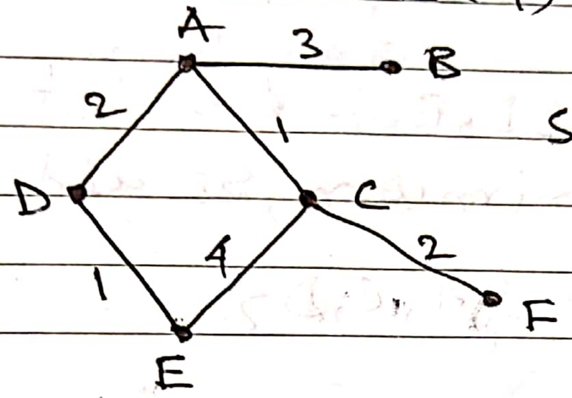
(3) Let w be the newest member of S.
For each $v \notin S$ & $v \in N_G(w)$ set $t(v) = \min\{t(v), t(w) + \alpha(vw)\}$
↳ what it was before

(4) Pick any $w \notin S$ with minimum $t(w)$ and add w to S.
Repeat step (3)

Solution: For each $v \in S$, $t(v)$ is the minimum weight of a path in G with weight function α.

Example:

$$\alpha : E(G) \longrightarrow \mathbb{N}$$



Source vertex: A

Step 1: $S = \phi$

Step 2: $t(A) = 0$ and $t(B) = t(C) = t(D) = t(E) = t(F) = \infty$

Now $S = \{A\}$

Step 3: $t(B) = \min\{\underbrace{\infty}_{t(B)\text{ earlier}}, \overbrace{0}^{t(A)} + \underbrace{3}_{\alpha(AB)}\} = 3$ — A is new member of S

$t(C) = \min\{\infty, 0+1\} = 1$

$t(D) = \min\{\infty, 0+2\} = 2$

Step 4: $C \notin S$ and $t(C) = 1$ is minimum so add C to S.

$\therefore S = \{A, C\}$

Repeating Step 3: New member in S is C. Neighbours of C

Date
Page

which does not belongs to S are E & F.

$t(E) = \min\{\infty, t(C) + \alpha(CE)\}$

$= \min\{\infty, 1 + 4\} = 5$

$t(F) = \min\{\infty, 1+2\} = 3$

Step 4: B, D, E, F $\notin$ S but $t(D) = 2$ is minimum, so add D to S.

$S = \{A, C, D\}$

Repeat Step 3: Now member of S is D. Neighbour of D but does not belongs to S is E.

$t(E) = \min(\cancel{5}, 2+1\} = 3$

Step 4: Add B to S

$S = \{A, C, D, B\}$

$\nexists$ any neighbour of B which does not belongs to S.

Add E to S.

$S = \{A, C, D, B, E\}$

(no change like previous)

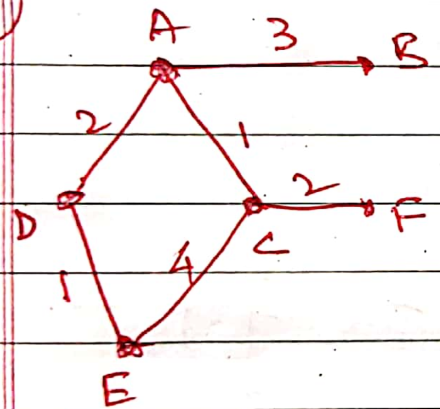
Add F to S

$S = \{A, C, D, B, E, F\}$

Now, $t(A) = 0$
$t(B) = 3$
$t(C) = 1$
$t(D) = 2$
$t(E) = 3$
$t(F) = 3$

Alternative process:

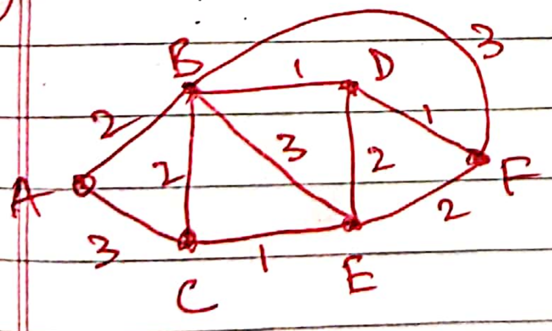
①



$S = \{A, C, D, B, E, F\}$

A 0
B ∞ 3 3 3
C $\cancel{\infty}$ 1
D ∞ 2 2
E ∞ ∞ 5 3 3
F ∞ ∞ 3 3 3 3

②



$S = \{A, B, C, D, E, F\}$

A 0
B ∞ 2
C ∞ 3 3
D ∞ ∞ 3 3
E ∞ ∞ 5 4 4
F ∞ ∞ 5 5 4 4

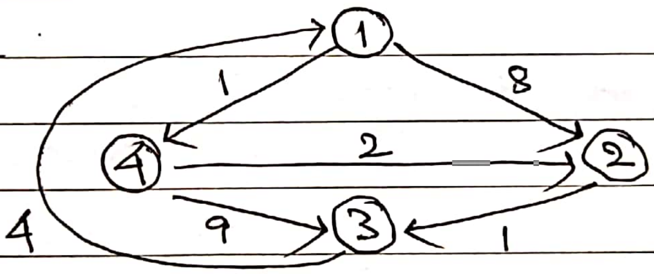
Floyd-Warshall algorithm:

* An algorithm to find solun of all Pairs Shortest Path Problem which gives the Shortest Path between every Pair of vertices of a given edge weighted graph.

* The algorithm is simple & easy to implement.

Algorithm:

(Ex1)



Initial distance matrix

$$D_0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & \infty & 1 \\ 2 & \infty & 0 & 1 & \infty \\ 3 & 4 & \infty & 0 & \infty \\ 4 & \infty & 2 & 9 & 0 \end{array}$$

$$D_1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & \infty & 1 \\ 2 & \infty & 0 & 1 & \infty \\ 3 & 4 & 12 & 0 & 5 \\ 4 & \infty & 2 & 9 & 0 \end{array}$$

$$D_2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 8 & 9 & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 0 & 5 \\ \infty & 2 & 3 & 0 \end{bmatrix} \end{array}$$

$$D_3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 8 & 9 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 12 & 0 & 5 \\ 7 & 2 & 3 & 0 \end{bmatrix} \end{array}$$

shortest distance matrix $D_4 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & 4 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 7 & 0 & 5 \\ 7 & 2 & 3 & 0 \end{bmatrix} \end{array}$

<u>NOTE:</u>

(i) Remove all the self loops and Parallel edges (Keeping the edge with lowest weight) from the graph.

(ii) For diagonal elements, value = 0.

(iii) For vertices having no direct edges betwn them, value = ∞.

Date
Page

Step 0: Let us consider the given graph is simple & has n vertices.

Step 1: Find the $n \times n$ initial distance matrix.
(diagonal will always be zero)

Step 2: To find the next $n \times n$ matrix D_i, $i = 1, 2, \ldots, n$.

(i) Write ith row & ith column as it is in the matrix D_{i-1}.

(ii) If $\exists$ any ∞ in the ith row or ith column, then write the corresponding row or column as it is in D_{i-1}.

(iii) To fill the other (p,q)th position, take the sum,

$$D_i(p,q) = D_{i-1}(p,i) + D_{i-1}(i,q) \quad p = q = 1, 2, \ldots, n \neq i$$

(iv) If $D_i(p,q) > D_{i-1}(p,q)$

then $D_{i-1}(p,q) = D_i(p,q)$.

(v) Diagonal will always be zero.

Step 3: Repeat Step 2 until D_n. → Answer

Date
Page

Time Complexity:

* Floyd-Warshall algorithm consists of three loops over all vertices.

$i = j$, $i \xrightarrow{?} j$, $i \not\to j$

* The loop in the middle consists of only operations of a constant complexity.

* Hence, the asymptotic complexity of FW algorithm is $O(n^3)$, where n is the no. of vertices in the graph.

Scanned with CamScanner