

Shortest Path Algorithm:

→ determines the minimum cost of the path from source to destination.

→ Cost of the path v_1, v_2, \dots, v_n is

$$\sum_{i=1}^{N-1} C_{i, i+1}$$

→ This is referred as weighted path length.

→ Unweighted path length → No of edges on the path namely $N-1$.

→ 2 types

① Single source shortest path
Unweighted shortest path
Dijkstra's Algorithm

② All pair shortest paths
Warshall Algorithm
Floyd's Algorithm.

Unweighted shortest path:

↳ i/p Graph $G=(V, E)$ and distinguished vertex s .

→ applied to both weighted & Unweighted Graph.

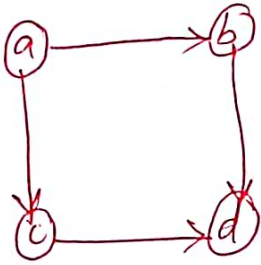
→ 3 pieces of info is maintained

Known

Distance Vertices (d_v) - Initially all vertices are ∞

Path (P_v)

eg.



step 1: Initialize table

V	Known	d_v	P_v
a	0	0	0
b	0	∞	0
c	0	∞	0
d	0	∞	0

source vertex 'a' is initially assigned a path length '0'.

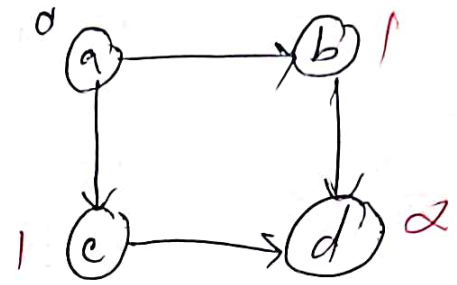
step 2:

Update adjacent vertices of 'a'

adjacent of 'a' is b, c

	a	is	dequeued
v	known	du	Pv
a	1	0	0
b	0	1	a
c	0	1	a
d	0	0	0

Enqueue : b, c

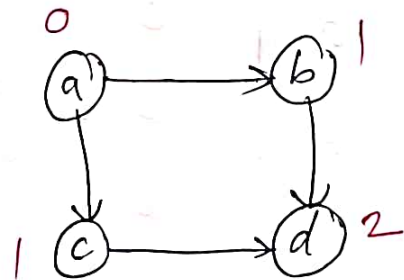


Step 3:

Next visit node 'b' & enqueue the unvisited adjacent of 'b'
 adjacent of 'b' is 'd'

	b	is	dequeued
v	known	du	Pv
a	1	0	0
b	1	1	a
c	0	1	a
d	0	2	b

Enqueue c, d



Step 4: Visit Node 'c'

	c	is	dequeued
v	known	du	Pv
a	1	0	0
b	1	1	a
c	1	1	a
d	0	2	b

Enqueue 'd'

step 5: D is dequeued Queue becomes empty.

V	known	d _v	P _v
a	1	0	0
b	1	1	a
c	1	1	a
d	1	2	b

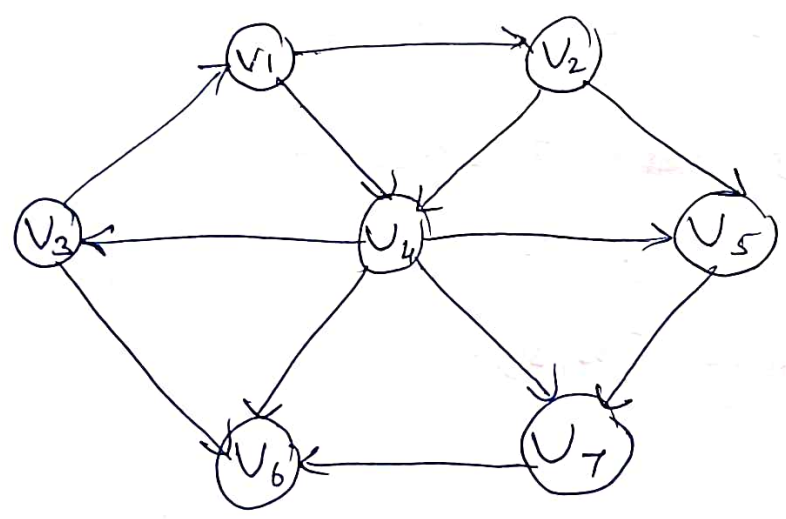
shortest path from source vertex 'a' is

a → b is 1

a → c is 1

a → d is 2.

Eg:-



Routine:

Void unweighted (Table T)

{

Queue Q;

Vertex v, w;

Q = create queue (Num Verten);

Make empty (Q);

Enqueue (s, Q);

while (! Is empty (Q))

{

v = dequeue (Q);

T[v]. known = True;

for each w adjacent to v

if (T[w]. dist == ∞)

{

T[w]. dist = T[v]. dist + 1;

T[w]. path = v;

Enqueue (w, Q);

}

}

dispose queue (Q);

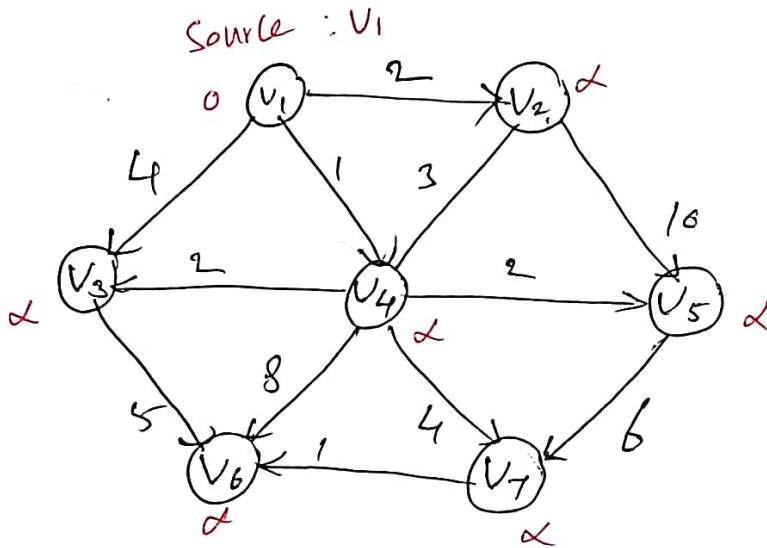
}

Dijkstra's Algorithm

Shortest path is known as Dijkstra's alg. General method to solve single source

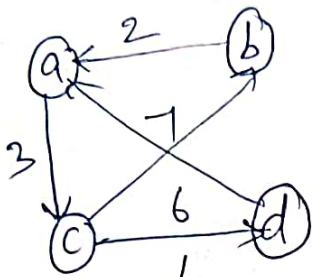
$$\text{check} - d_w = d_v + C_{vw}$$

Eg:



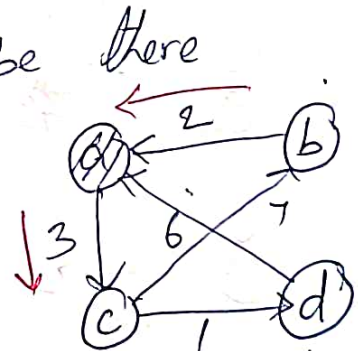
Floyd's Algorithm:

Eg-1



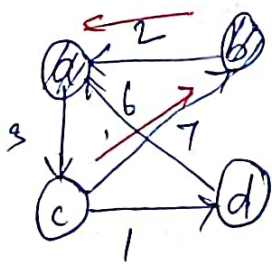
$$D^{[0]} = \begin{matrix} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} 0 & 2 & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix} \end{matrix}$$

$D^{[1]}$ - 1 intermediate node will be there

$$\begin{matrix} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \boxed{5} & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \boxed{9} & 0 \end{bmatrix} \end{matrix}$$


$d \rightarrow a \rightarrow c = 9$
 $b \rightarrow a \rightarrow c = 5$

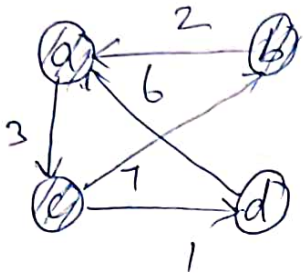
$D^{[2]}$ - 2 intermediate nodes



$$\begin{matrix} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \boxed{9} & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix} \end{matrix}$$

$c \rightarrow b \rightarrow a = 9$

D^[3] - 3 intermediate nodes [a, b, c]



	a	b	c	d
a	0	10	3	4
b	2	0	5	6
c	9	7	0	1
d	6	16	9	0

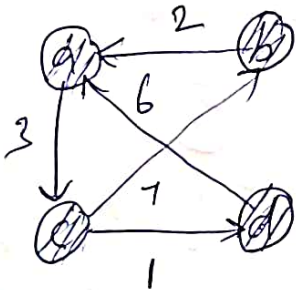
$$a \rightarrow c \rightarrow b = 10$$

$$a \rightarrow c \rightarrow d = 4$$

$$b \rightarrow a \rightarrow c \rightarrow d = 6$$

$$d \xrightarrow{6} a \xrightarrow{3} c \xrightarrow{7} b = 16$$

D^[4] - 4 intermediate nodes [a, b, c, d]



$$c \rightarrow d \rightarrow a = 7$$

	a	b	c	d
a	0	10	3	4
b	2	0	5	6
c	7	9	0	1
d	6	16	9	0