Some notations:

Shortest paths in graphs: The graph G has n vertices and a distance associated with each edge of the graph G(such a graph is often called a network). The representation of the network will be as a distance matrix D.
The **distance matrix** $D = (d_{ij})$ where, $d_{ij} = 0$, if i = j.
$d_{ij} = \infty$, if i is not joined to j by an edge.
$d_{ij} =$ distance associated with an edge from i to j, if i is joined to j by an edge.

We shall find the shortest distance between the vertices of a graph G using **Dijkstra′s algorithm.**

Let us define two sets K and U, where K consists of those vertices which have been fully investigated and between which the best path is known, and U of those vertices which have not yet been processed. Clearly, every vertex belongs to either K or U but not both. Let a vertex r be selected from which we shall find the shortest paths to all the other vertices of the network. Let the array $bestd(i)$ hold the length of the shortest path so far formed from r to vertex i, and another array $tree(i)$ the next vertex to i on the current shortest path.

Dijkstra′s algorithm:

Step 1: Intialise $K = \{r\}$, U=$\{$ all other vertices of G except $r\}$. Set bestd(i)= d_{ri} and tree(i)= r.

Step 2: Find the vertex s in U which has the minimum value of bestd. Remove s from U and put it in K.

Step 3: For each vertex u in U, find bestd(s) $+$ d_{su} and if it is less than bestd(u) replace bestd(u) by this new value and let tree(u)= s.(a shorter path to u has been found by going via vertex s.)

Step 4: If U contains only one vertex then stop the process or else go to step 2. The array bestd(i) contains the length of shortest path from r to i.

Dijkstra's Algorithm (Explicit)

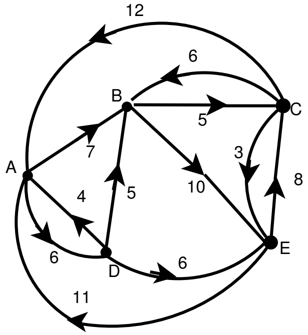
Step 1: Initialize $K = \{r\}$, $U = V(G) \setminus \{r\}$.
For each i: set $bestd(i) = d_{ri}$,
$tree(i) = r$.

Step 2: Find $s \in U$ which has the minimum value of $bestd$.
$U = U \setminus \{s\}$,
$K = K \cup \{s\}$.

Step 3: For each $u \in U$:
If $bestd(s) + d_{su} < bestd(u)$:
$bestd(u) = bestd(s) + d_{su}$
$tree(u) = s$.

Step 4: If $|U| = 1$: STOP.
Else: Go back to **Step 2**.

Example: Implement Dijkstra′s algorithm to find shortest path from the vertex B to all other vertices of following graph G.



$$D(G) = \begin{bmatrix} 0 & 7 & \infty & 6 & \infty \\ \infty & 0 & 5 & \infty & 10 \\ 12 & 6 & 0 & \infty & 3 \\ 4 & 5 & \infty & 0 & 6 \\ 11 & \infty & 8 & \infty & 0 \end{bmatrix}$$

Figure: Graph G and its distance matrix $D(G)$

(1) $K = \{B\}$, $U = \{A, C, D, E\}$.

Vertex in U	A	$\boxed{\text{C}}$	D	E
bestd	∞	5	∞	10
tree	B	B	B	B

(2) min $bestd = 5$; $s = C$.
$U = \{A, D, E\}$, $K = \{B, C\}$.

(3) $u = A$: $bestd(C) + d_{CA} = 5 + 12 = 17 < \infty = bestd(A)$.
Update: $bestd(A) = 17$, $tree(A) = C$.
$u = D$: $bestd(C) + d_{CD} = 5 + \infty \not< \infty = bestd(D)$.
$u = E$: $bestd(C) + d_{CE} = 5 + 3 = 8 < 10 = bestd(E)$.
Update: $bestd(E) = 8$, $tree(E) = C$.

Vertex in U	A	D	$\boxed{\text{E}}$
bestd	17	∞	8
tree	C	B	C

(4) min $bestd = 8$; $s = E$.
$U = \{A, D\}$, $K = \{B, C, E\}$.

(5) $u = A$: $bestd(E) + d_{EA} = 8 + 11 = 19 \not< 17 = bestd(A)$.
$u = D$: $bestd(E) + d_{ED} = 8 + \infty \not< \infty = bestd(D)$.

Vertex in U	$\boxed{\text{A}}$	D
bestd	17	∞
tree	C	B

(6) min $bestd = 17$; $s = A$.
$U = \{D\}$, $K = \{A, B, C, E\}$.

(7) $u = D$: $bestd(A) + d_{AD} = 17 + 6 = 23 < \infty = bestd(D)$.
Update: $bestd(D) = 23$, $tree(D) = A$.

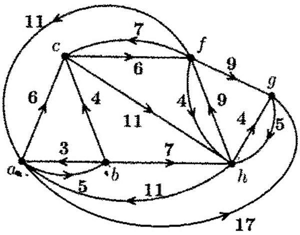
Vertex in U	D
bestd	23
tree	A

(8) $|U| = 1$. Hence STOP.

The shortest path from B to all the vertices of G is given by

Vertex	A	B	C	D	E
bestd	17	0	5	23	8
tree	C	B	B	A	C

Example: Implement Dijkstra′s algorithm to find shortest path from c to all other vertices of the following network.



$$D(G) = \begin{bmatrix} 0 & 5 & 6 & \infty & 17 & \infty \\ 3 & 0 & 4 & \infty & \infty & 7 \\ \infty & \infty & 0 & 6 & \infty & 11 \\ 11 & \infty & 7 & 0 & 9 & 4 \\ \infty & \infty & \infty & \infty & 0 & 5 \\ 11 & \infty & \infty & 9 & 4 & 0 \end{bmatrix}$$

Figure: Graph G and its distance matrix $D(G)$

(1) $K = \{c\}$, $U = \{a, b, f, g, h\}$.

Vertex in U	a	b	$\boxed{\text{f}}$	g	h
bestd	∞	∞	6	∞	11
tree	c	c	c	c	c

(2) min $bestd = 6$; $s = f$.
$U = \{a, b, g, h\}$, $K = \{c, f\}$.

(3) $u = a$: $bestd(f) + d_{fa} = 6 + 11 = 17 < \infty = bestd(a)$.
Update: $bestd(a) = 17$, $tree(a) = f$.
$u = b$: $bestd(f) + d_{fb} = 6 + \infty \not< \infty = bestd(b)$.
$u = g$: $bestd(f) + d_{fg} = 6 + 9 = 15 < \infty = bestd(g)$.
Update: $bestd(g) = 15$, $tree(g) = f$.
$u = h$: $bestd(f) + d_{fh} = 6 + 4 = 10 < 11 = bestd(h)$.
Update: $bestd(h) = 11$, $tree(h) = f$.

Vertex in U	a	b	g	$\boxed{\text{h}}$
bestd	17	∞	15	10
tree	f	c	f	f

(4) min $bestd = 10$; $s = h$.
$U = \{a, b, g\}$, $K = \{c, f, h\}$.

(5) $u = a$: $bestd(h) + d_{ha} = 10 + 11 = 21 \not< 17 = bestd(a)$.
$u = b$: $bestd(h) + d_{hb} = 10 + \infty \not< \infty = bestd(b)$.
$u = g$: $bestd(h) + d_{hg} = 10 + 4 = 14 < 15 = bestd(g)$.
Update: $bestd(g) = 14$, $tree(g) = h$.

Vertex in U	a	b	$\boxed{\text{g}}$
bestd	17	∞	14
tree	f	c	h

(6) min $bestd = 14$; $s = g$.
$U = \{a, b\}$, $K = \{c, f, g, h\}$.

(7) $u = a$: $bestd(g) + d_{ga} = 14 + \infty \not< 17 = bestd(a)$.
$u = b$: $bestd(g) + d_{gb} = 14 + \infty \not< \infty = bestd(b)$.

Vertex in U	$\boxed{\text{a}}$	b
bestd	17	∞
tree	f	c

(8) min $bestd = 17$; $s = a$.
$U = \{b\}$, $K = \{a, c, f, g, h\}$.

(9) $u = b$: $bestd(a) + d_{ab} = 17 + 5 = 22 < \infty = bestd(b)$.
Update: $bestd(b) = 22$, $tree(b) = a$.

Vertex in U	b
bestd	22
tree	a

(10) $\mid U \mid = 1$. Hence STOP.

The shortest path from c to all the vertices of G is given by

Vertex	a	b	c	f	g	h
bestd	17	22	0	6	14	10
tree	f	a	c	c	h	f

References

[1] Frank Harary, *Graph Theory*, Addison-Wesley Publishing Company, Inc, 1969.

[2] E S Page and L B Wilson, *An introduction to Computational Combinatorics*, Cambridge University Press, Inc, 1979.

[3] Narsingh Deo, Graph theory with Application to Engineering and Computer Science, PHI, 1987.