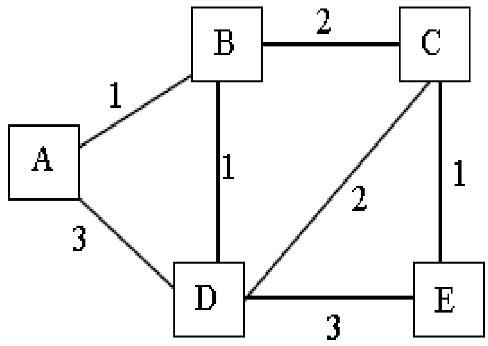
Shortest Path Routing Example
(Building the Routing Table)

Routing algorithms example

Consider the following network topology. The boxes indicate nodes and the integers next to the arcs are the link cost. All links are bi-directional.



1. Use Dijkstra's algorithm to compute the least cost routes from node A to all other nodes. Show your working in a set of tables with columns "Step" (number of times in the loop), N, and "D(X), p(X)" for all X not equal to A.

Dijkstra's algorithm is a centralised algorithm, so all the topology is known.

The routing table looks like this:

Step	N	D(B), p(B)	D(C), p(C)	D(D), p(D)	D(E), p(E)
0	{A}	**1,A**	∞	**3,A**	∞
1					

Step is the iteration number of the algorithm.
c(i,j): link cost from node i to j. cost infinite if not direct neighbours
D(v): current value of cost of path from source to destination V
p(v): predecessor node along path from source to v, that is next v
N: set of nodes whose least cost path definitively known

Infinite entries mean no route has yet been found. Bold entries are new.

So here's the answer….

Step	N	D(B), p (B)	D(C), p (C)	D(D), p (D)	D(E), p (E)	Comments
0	{A}	**1,A**	∞	**3,A**	∞	Start at A
1	{A,B}	1,A	**3,B**	**2,B**	∞	AB is least cost, so add B to N. D(B) and p(B) will not change hereafter. Still can't get to E from any node in N.
2	{A,B,D}		3,B	2,B	**5,D**	ABD is least cost so add D to N. Can now see E (via ABDE). Cost of C doesn't change.
3	{A,B,D,C}		3,B		**4,C**	ABC is least cost so add C to N. Cost of ABCE is less than ABDE, so change.
4	{A,B,D,C,E}				4,C	Add E to N. All nodes now in N so end loop.