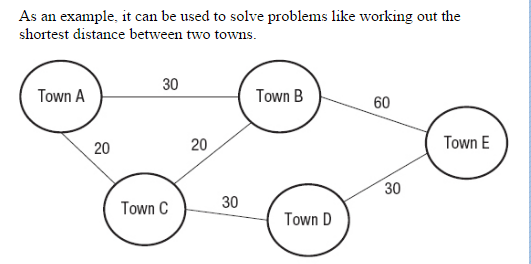
**Dijkstra’s shortest path algorithm**

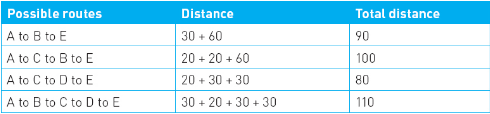
Dijkstra’s shortest path algorithm attempts to find the shortest distance to travel from one node to another.

This will only work with a weighted graph as there has to be positive values to show the distance between nodes (the length of the edges).



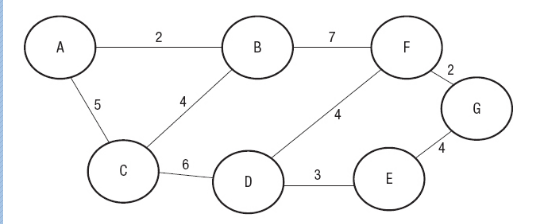
Destination node

Single Source



**Dijkstra’s shortest path step by step**

1. Start from the first vertex.
2. Find the weight for each edge between the first vertex and adjacent vertices.
3. Move on to the next vertex and repeat the process, adding the weights of each route. For example, A to B may be shorter than A to C, however A to C could be shorter than A to B to C.
4. Repeat this process until the destination vertex is reached.







This just means that there is no edge between A and G





Even though the weight of the edge between C and D is 6, we keep track of the accumulated weights, so it is 5+6 (A to C to D)

The edge between A and B has a weight of 2, the small letter next to the number shows the vertex that we came from.

There is no weight between any vertex and itself

**Tracing Dijkstra’s algorithm (above)**

A is the start vertex so we check the distance between A and its adjacent vertices, which are B and C.

The edge between A and B is shorter than A and C, so we move onto B next.

When we fill a value into the table, we keep it in the next row unless a smaller value is found to replace it, for example we keep 5A in row B because B to C is 4+2 which is larger than A to C which is 5. We also record the distance from B to F as 9, because that is the total weight when traveling from A to B to F.

The nearest vertex to B is C, so we move to that next.

We can see that the accumulated weight of going from C to D is 11, however the same weight applies from going from A to F, which is closer to G, therefore we record the weight of C to D, but move to F.

The total weight of A to B to F to G is 11, therefore this is the shortest path.

**Uses of Dijkstra’s shortest path algorithm**

Geographic information systems (GIS) where vertices (nodes) are geographical locations and edges show distance.

Telephone and computer network planning where the vertices are the individual devices on the network and the edges could be physical distance or a measurement of network capability.

Logistics: if there is a large network of delivery vehicles for example, the optimum routes can be found.