Dijkstra's algorithm

For each vertex (or node) in a graph, Dijkstra's algorithm finds the shortest path from the source node to that vertex. Figure 1. shows a graph with vertices A - F and edges with distances of 10,15,12,1,2 and 5.

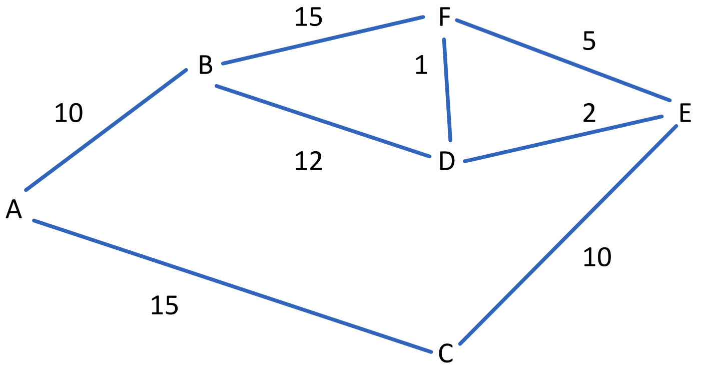


Figure 1. Graph in its initial state before the algorithm is applied.

When the algorithm is finished, the graph will show the shortest paths:

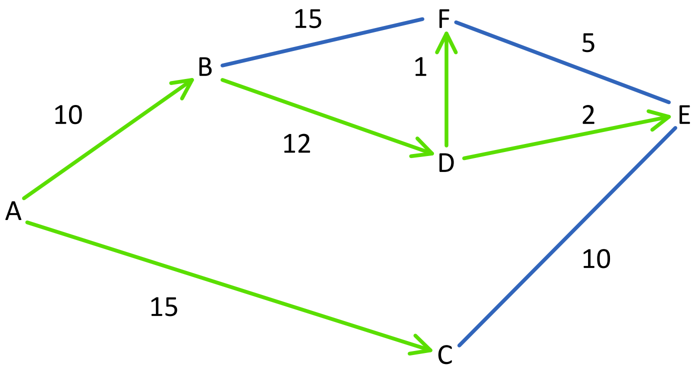
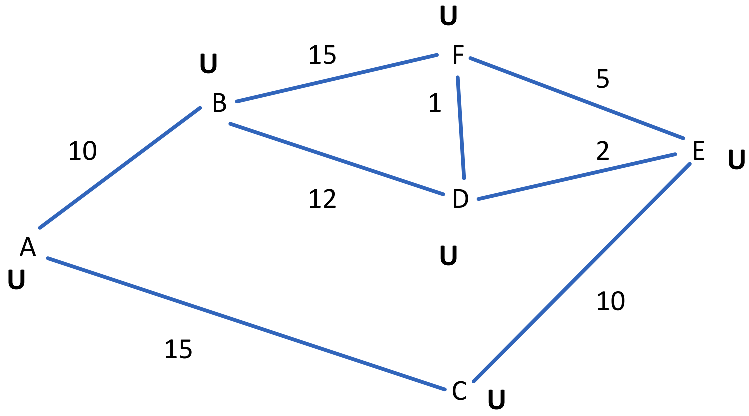


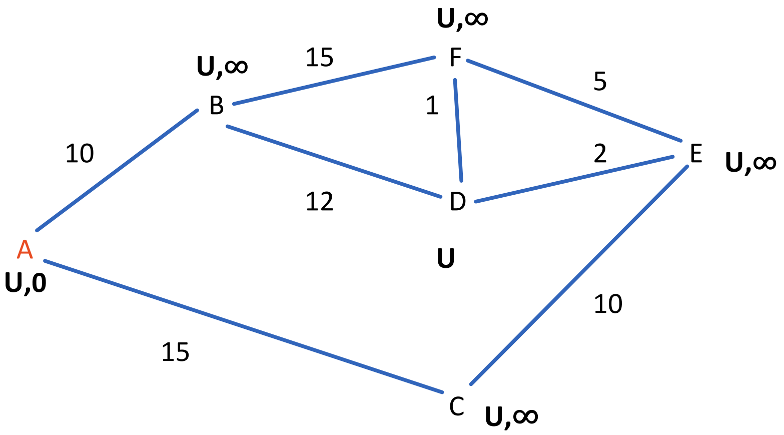
Figure 2. Graph with shortest paths from A to all other vertices.

Running the algorithm

Iteration1.STEP1 - Set all the nodes to unvisited, marking them with a U



Iteration1.STEP2 - Assign every node a temporary distance - set it to zero for our source node, A, and set all other nodes to infinity. Set our current node to A.



Iteration1.STEP3 - For each unvisited node attached to the current node, repeat this process:

1) Calculate the temporary distance for the unvisited node:

TEMPORARY_DISTANCE = CURRENT_NODE_TEMPORARY_DISTANCE + EDGE_TO_UNVISITED_NEIGHBOUR

2) Compare TEMPORARY_DISTANCE to the current temporary distance on the unvisited neighbour and see which is the smaller value

3) Update the temporary distance on the unvisited node with the smaller of the two values