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.
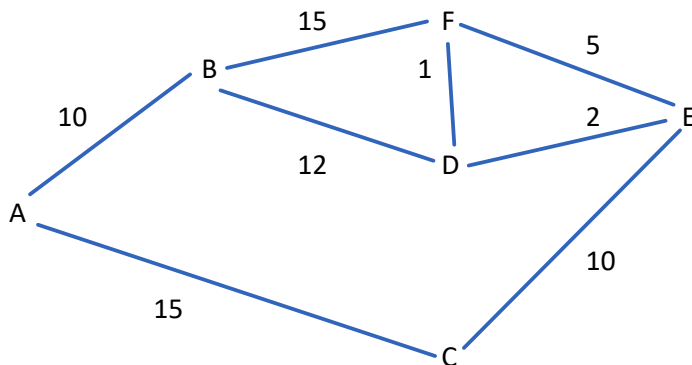

Figure 1. Graph in its initial state before the algorithm is applied.

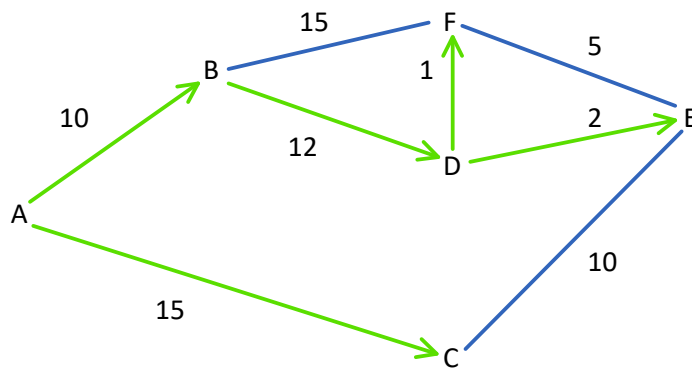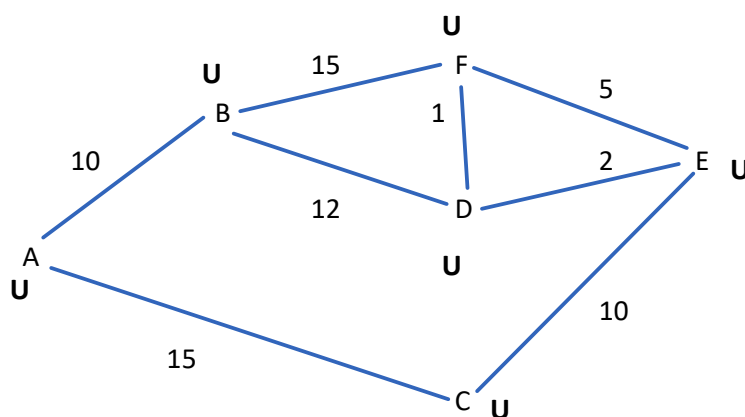When the algorithm is finished, the graph will show the shortest paths:

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.

U,∞
U,∞   15   F
B   1   5
10   2   E   U,∞
              12   D
A   U
U,0   10
         15
              C   U,∞
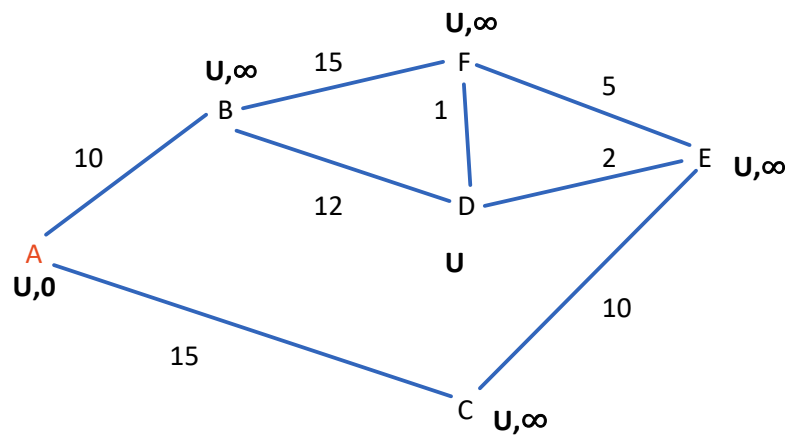
Iteration1.STEP3 - For each unvisited node directly attached to the current node, calculate the new temporary distance (Figures 3a - 3c).

U,∞
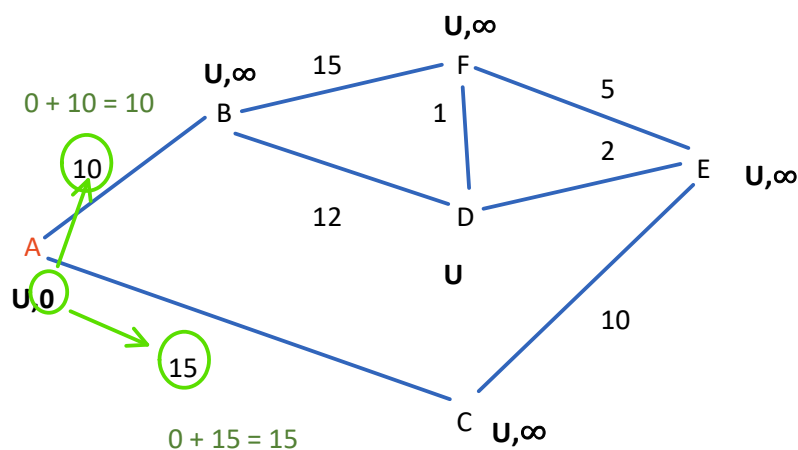U,∞   15   F
0 + 10 = 10   B   1   5
                         2   E   U,∞
10
              12   D
A   U
U,0   10
         15
0 + 15 = 15   C   U,∞

Figure 3a. Add the temporary distance on the current node to the edges of the unvisited nodes.

10 < ∞   U,∞
U,∞   15   F
0 + 10 = 10   B   1   5
                         2   E   U,∞
10
              12   D
A   U
U,0   10
         15
0 + 15 = 15   C   U,∞
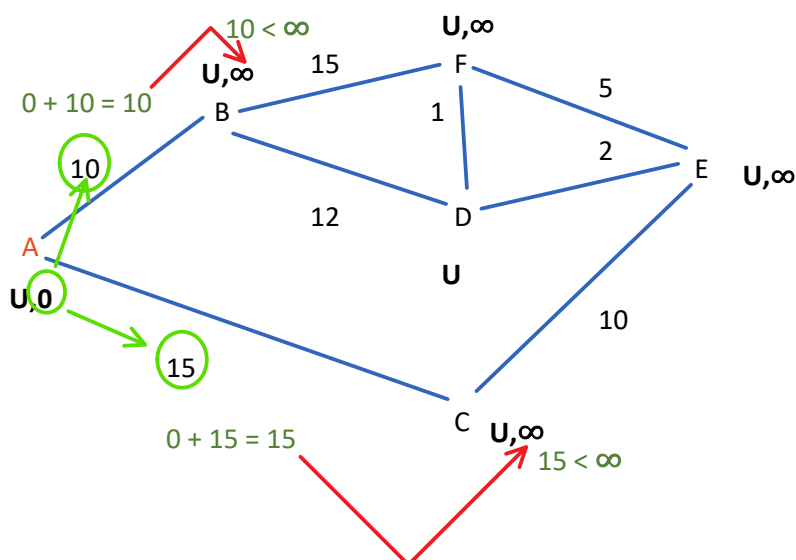   15 < ∞

Figure 3b. Compare the new temporary distance with the temporary distance on the unvisited node. For each node, record the value that is smaller.
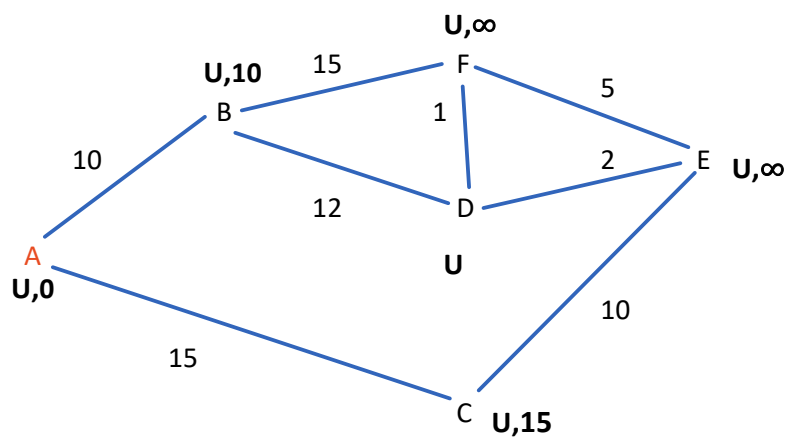
Figure 3c. Update the temporary distance on the unvisited node with the smaller value.