Dijkstra's Shortest Path Algorithm Notes

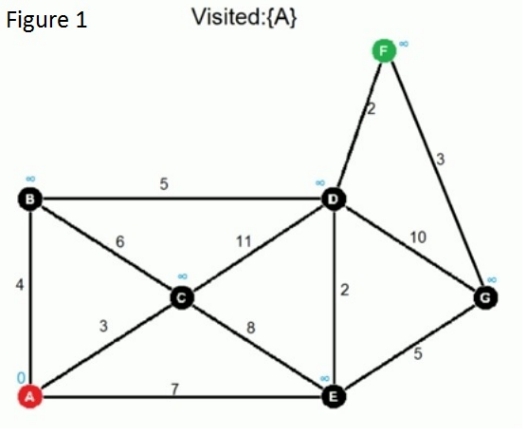
Shortest Path Algorithm

- An algorithm run on a weighted graph
- Starts with an initial node and a goal node
- Finds the least cost path to the goal node

Algorithm

- Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes (Figure 1)

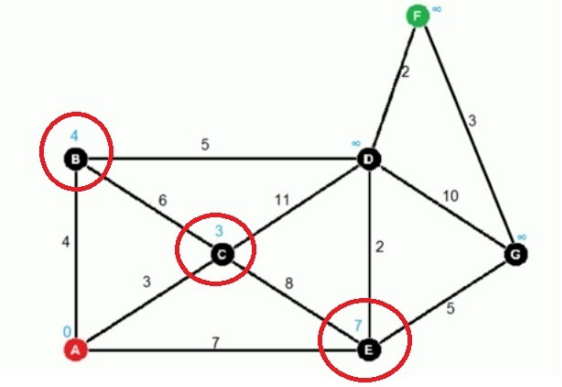
Figure 1 Visited:{A}



- Keep a set of visited nodes. This set starts with just the initial node

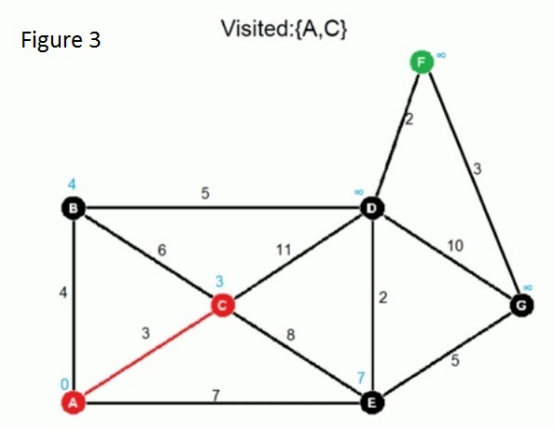
- For the current node, consider all of its unvisited neighbors and calculate (distance to the current node) + (distance from current node to the neighbor). If this is less than their current tentative distance, replace it with this new value (Figure 2).

Figure 2 Visited:{A}



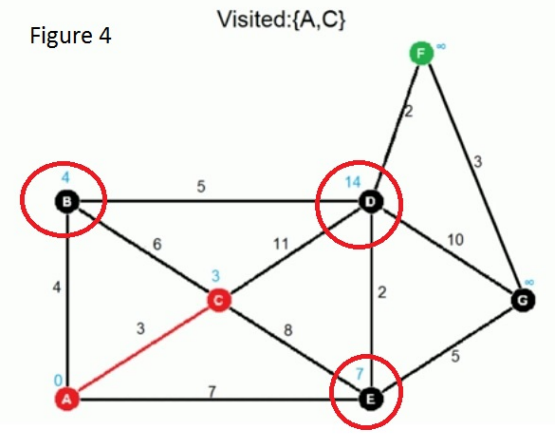
- When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the unvisited set (Figure 3)

Figure 3 Visited:{A,C}



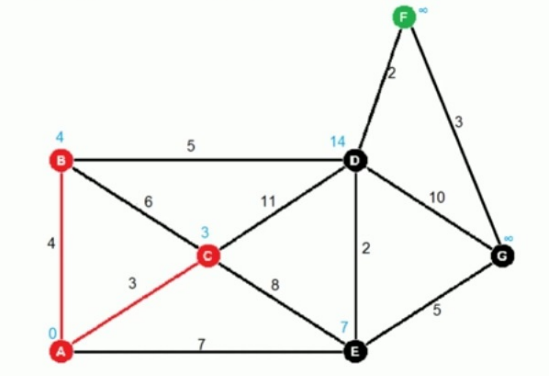
- After the first node is established, we will find the next shortest destination

Figure 4 Visited:{A,C}

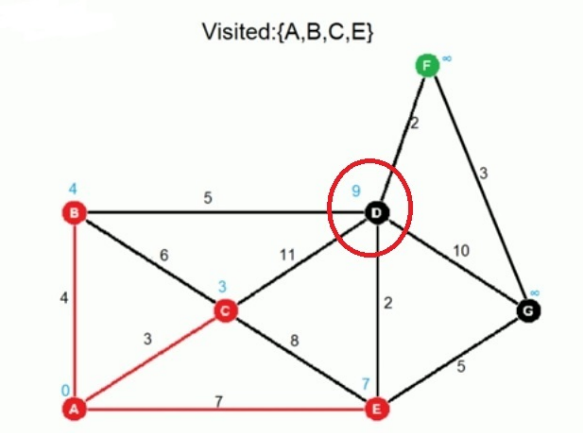


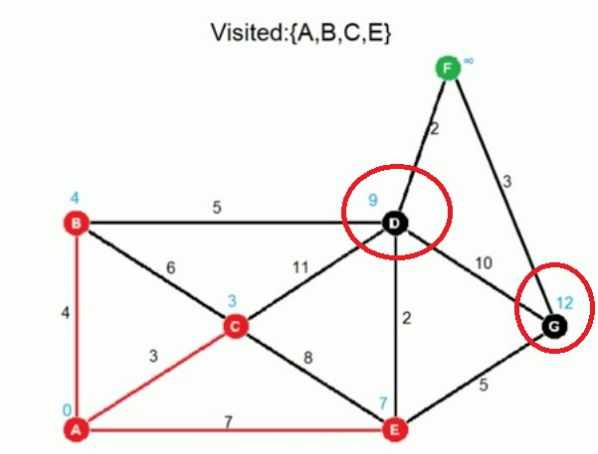
- The next 3 nodes are 4, 14, and 7 respectively (Figure 4)
- Since 4 is the shortest distance we take node B off the list (Figure 5)

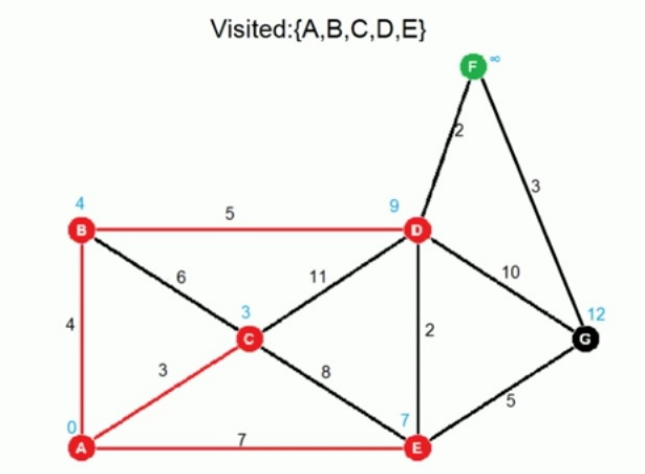
Figure 5 Visited:{A,B,C}

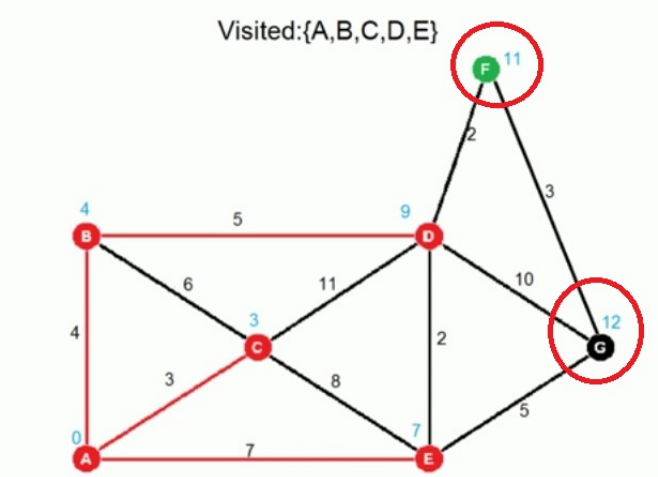


- Repeat previous steps of finding cumulative shortest distance

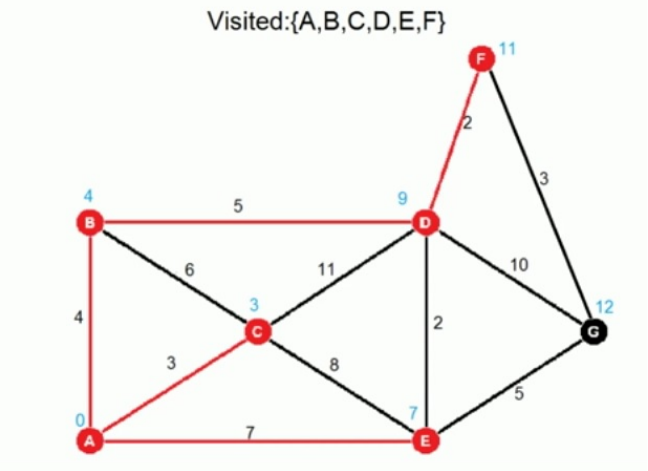








- If the destination node has been marked visited, the algorithm has finished (Figure 6)



- Set the unvisited node marked with the smallest algorithm has finished