HW7 - Dijkstra Algorithm

Ronan Wallace

May 2019

7) Dijkstra's Algorithm

Show what Dijkstra's algorithm would do to compute the shortest paths from node A to all the other nodes in the graph (see pictures below)

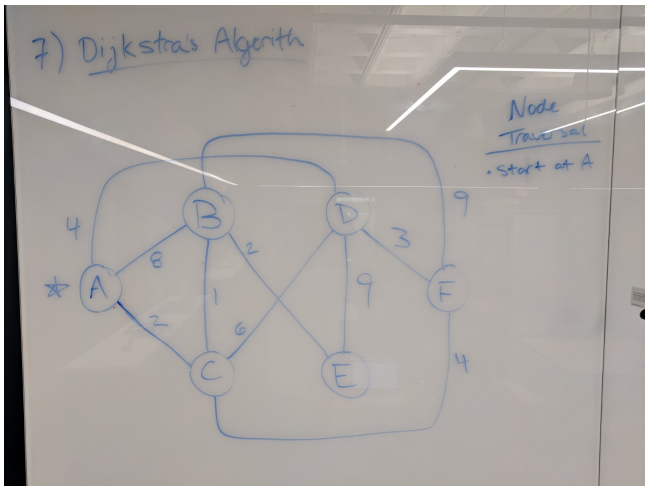

Figure 1: Plot graph. Choose node A and look at which adjacent edge has the least amount of weight to it.

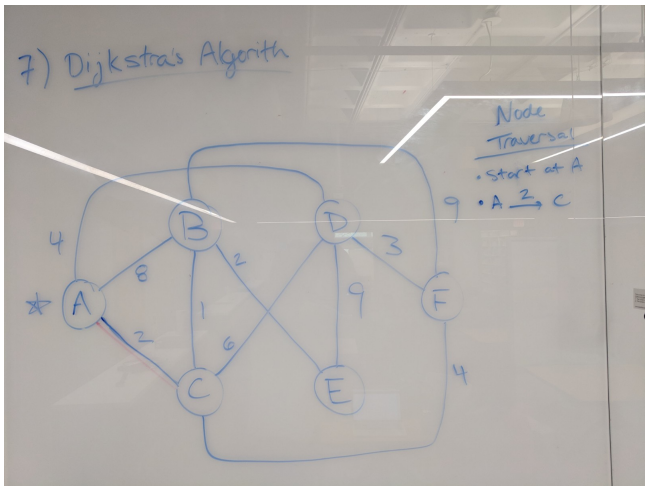

Figure 2: The least weighted edge is A $\rightarrow$ C with a weight of 2. Mark the edge and add the traversal to the traversal list (to keep track). Begin checking for the next least weighted edge to traverse from node C.

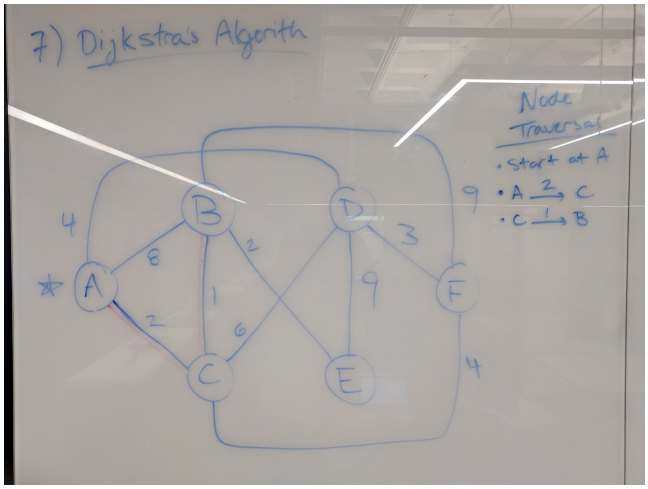

Figure 3: The least weighted edge is C → B with a weight of 1. Mark the edge and add the traversal to the traversal list. Begin checking for the next least weighted edge to traverse from node B.

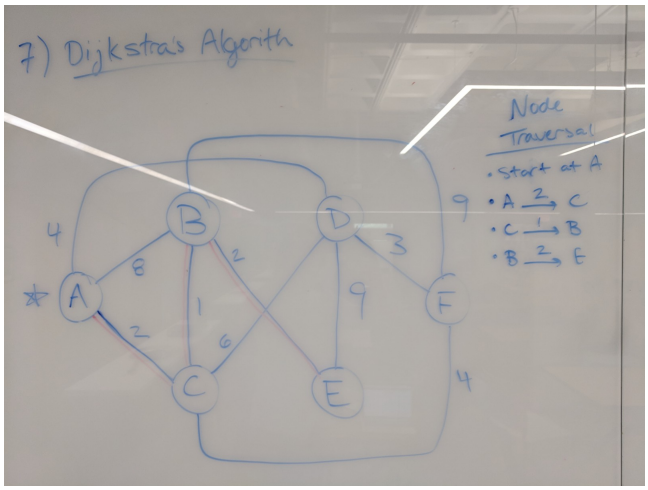

Figure 4: The least weighted edge is B → E with a weight of 2. This edge is chosen because although B → C has the least weighted edge, C has already been visited along with the edge having already been traversed. Mark the edge and add the traversal to the traversal list. Begin checking for the next least weighted edge to traverse from node E.

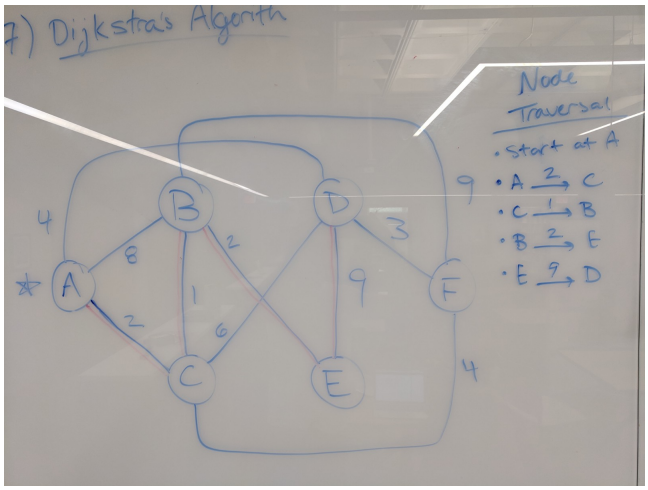

Figure 5: The least weighted (and only) edge is E → D with a weight of 9. Mark the edge and add the traversal to the traversal list. Begin checking for the next least weighted edge to traverse from node D.

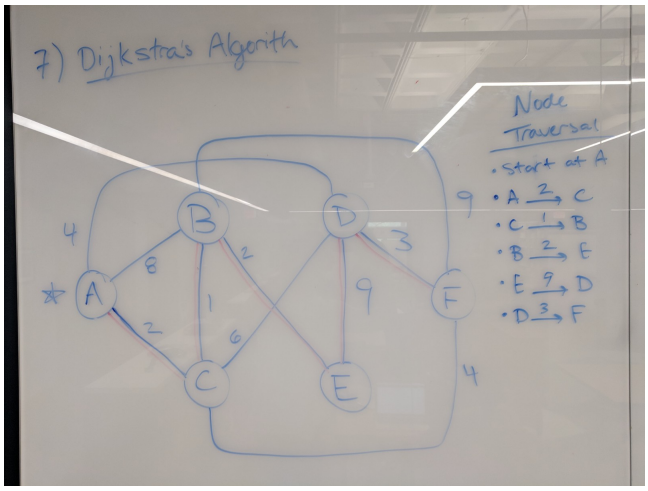

Figure 6: The least weighted edge is D → F with a weight of 3. Mark the edge and add the traversal to the traversal list. Begin checking for the next least weighted edge to traverse from node F.

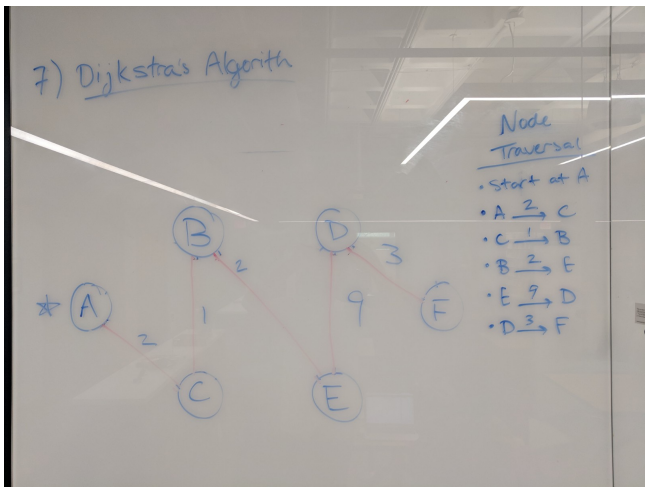

Figure 7: At this point, all nodes have been traversed and the shortest path from node A to all other nodes has been found.