Prim's Algorithm

Algorithm that is guaranteed to find a minimum spanning tree in a graph

1. **Pick a vertex in the network randomly, add that vertex to the spanning tree**

2. Examine all edges connected from a vertex in the spanning tree to a vertex NOT in the spanning tree

Of those edges, choose the one with the lowest (minimum) cost and add its attached vertex to the spanning tree

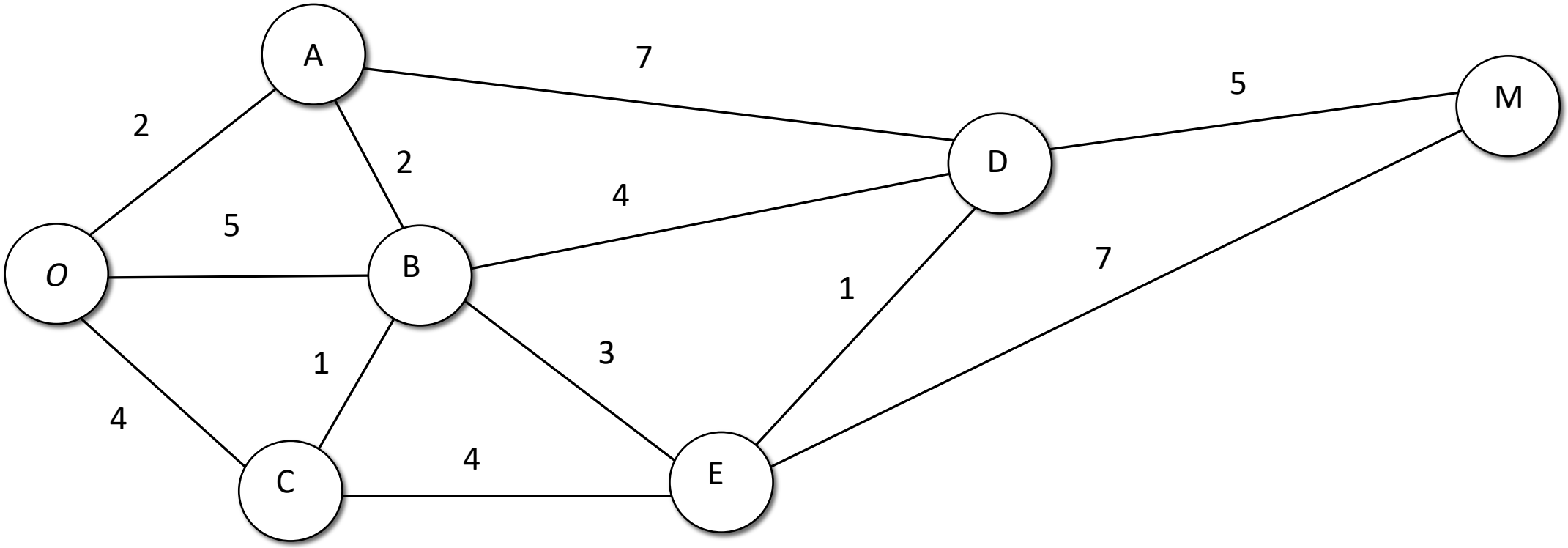
Break ties arbitrarily

3. Are all vertices attached to the spanning tree?

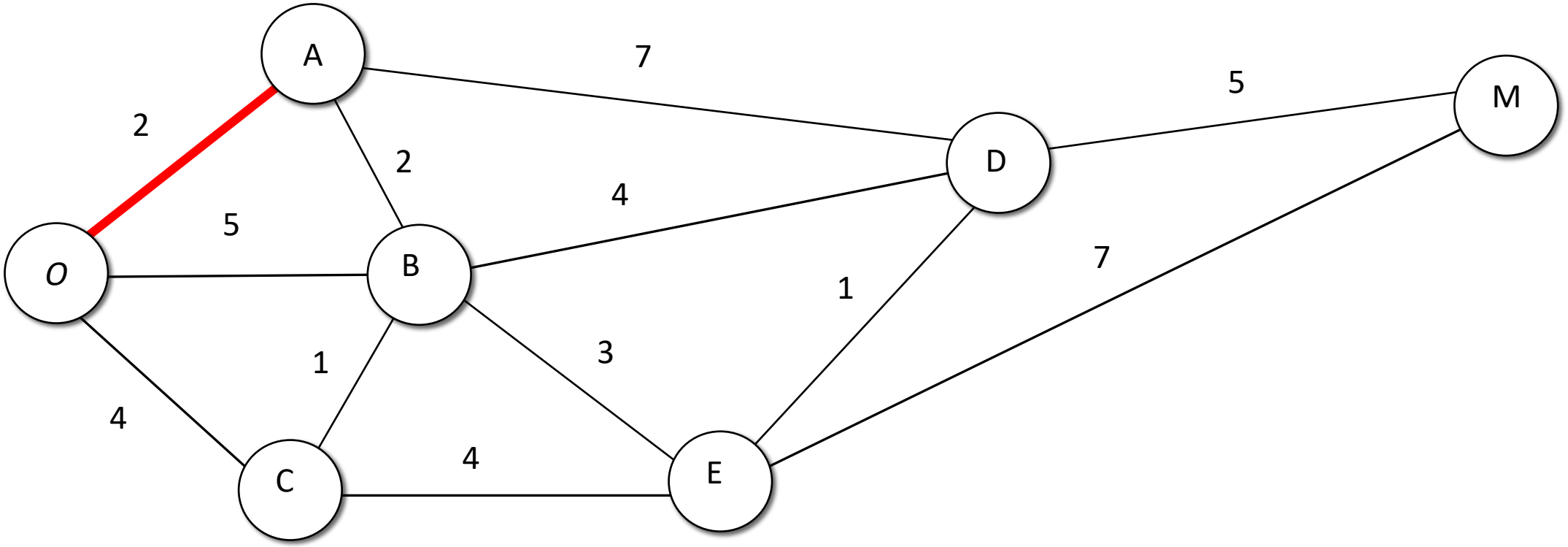
If yes, stop

If no, go to step 2.

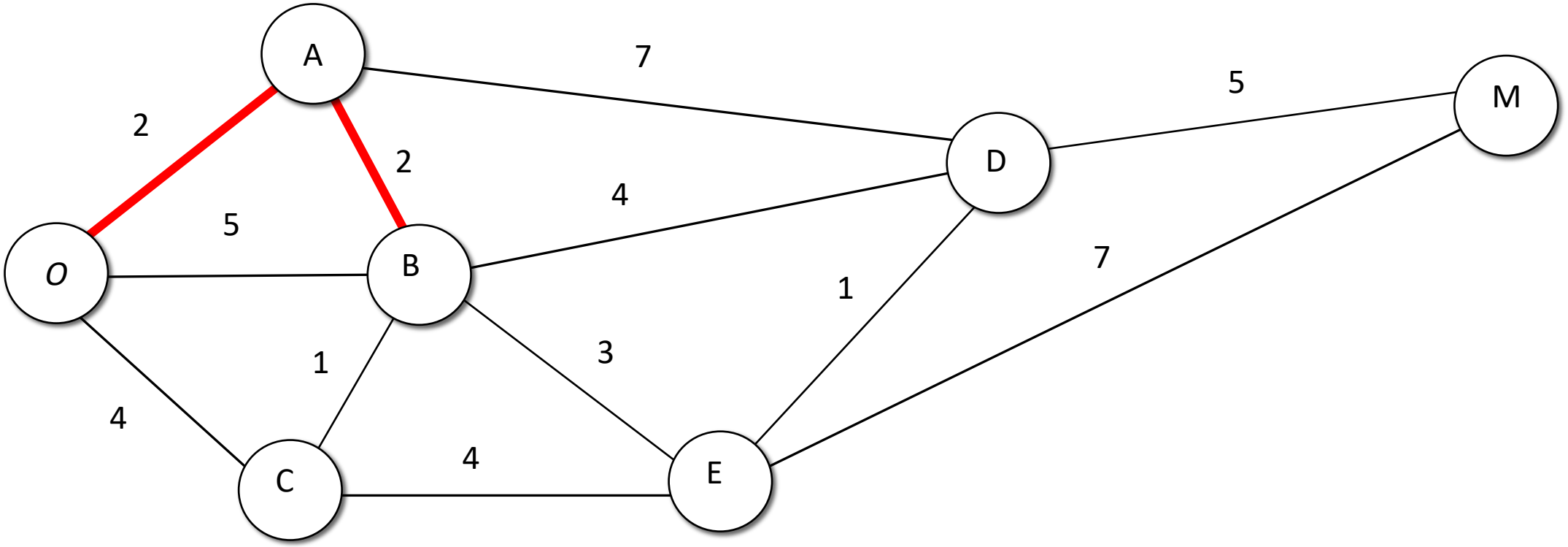
Minimum Spanning Tree – Start at O



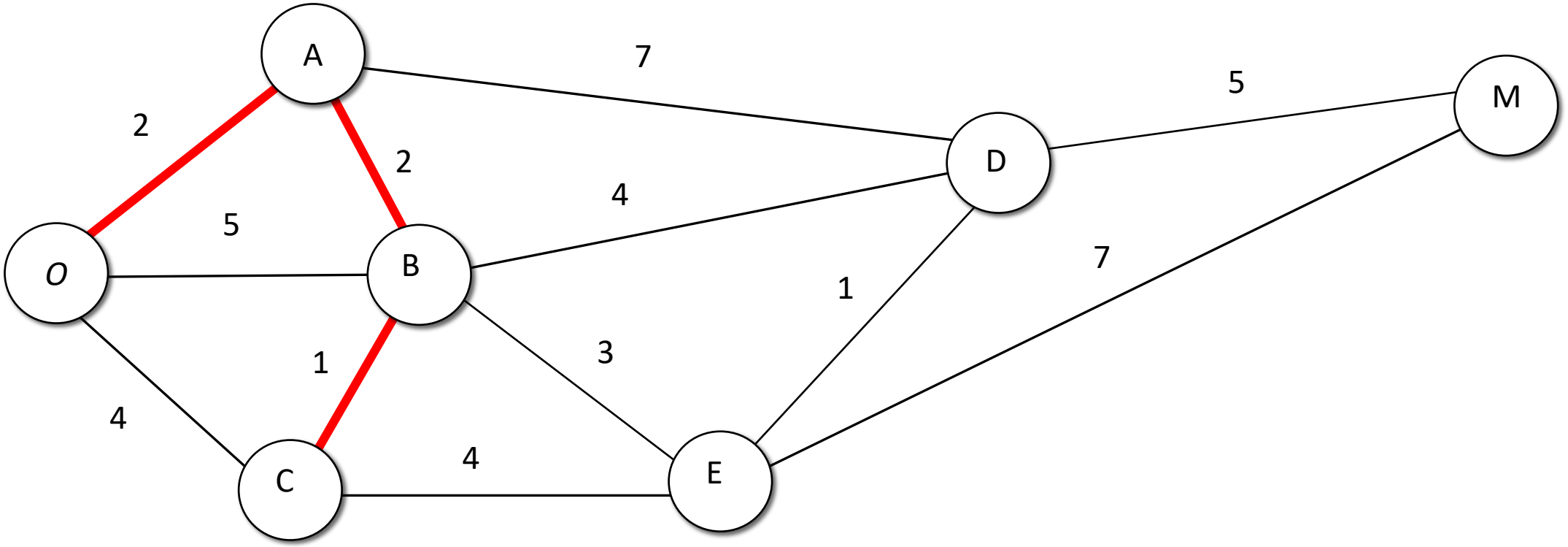
Minimum Spanning Tree – Start at O



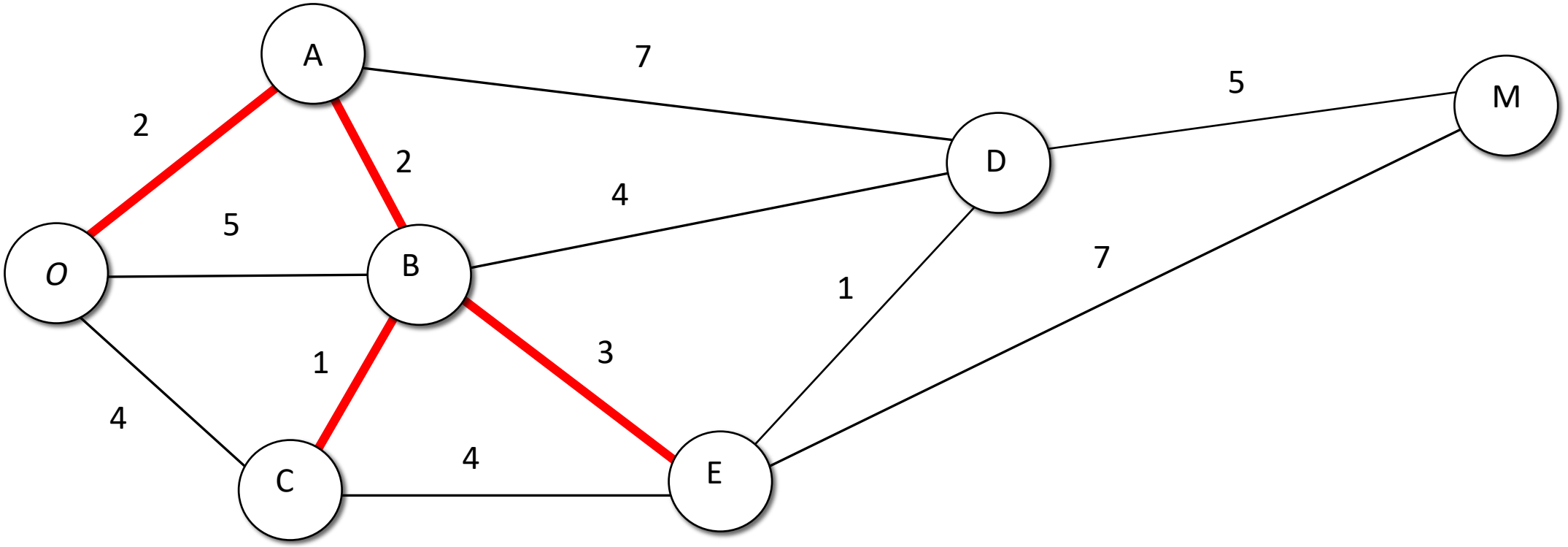
Minimum Spanning Tree – Start at O



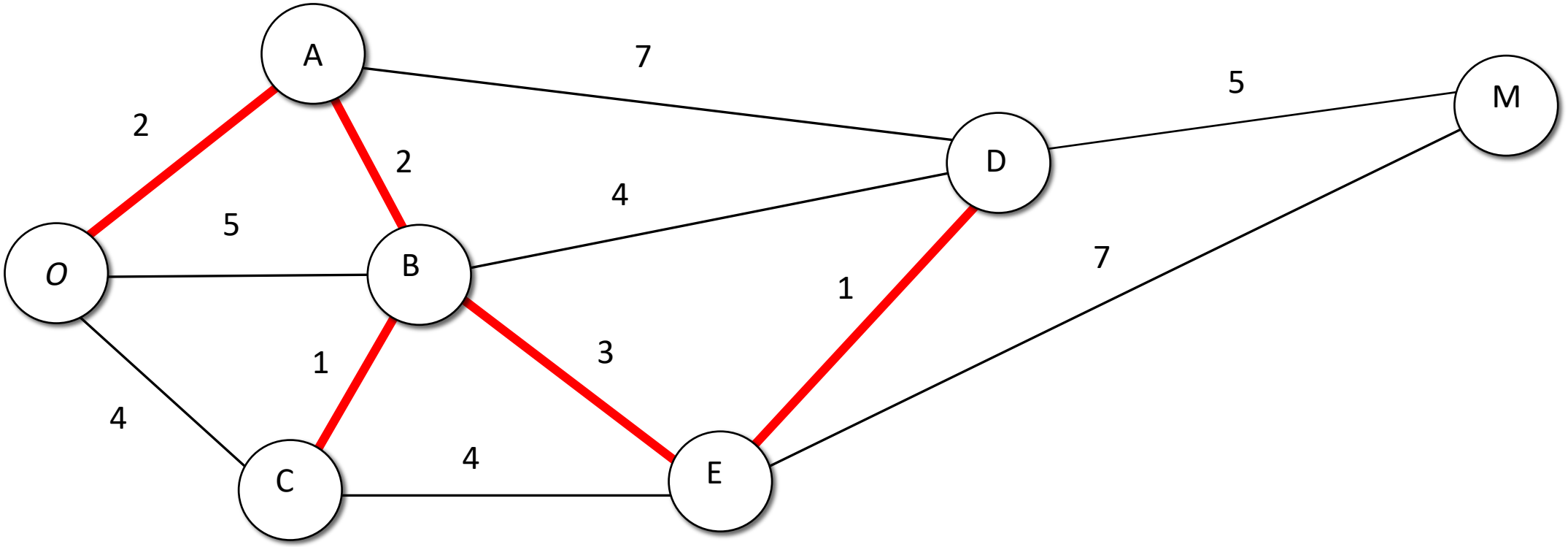
Minimum Spanning Tree – Start at O



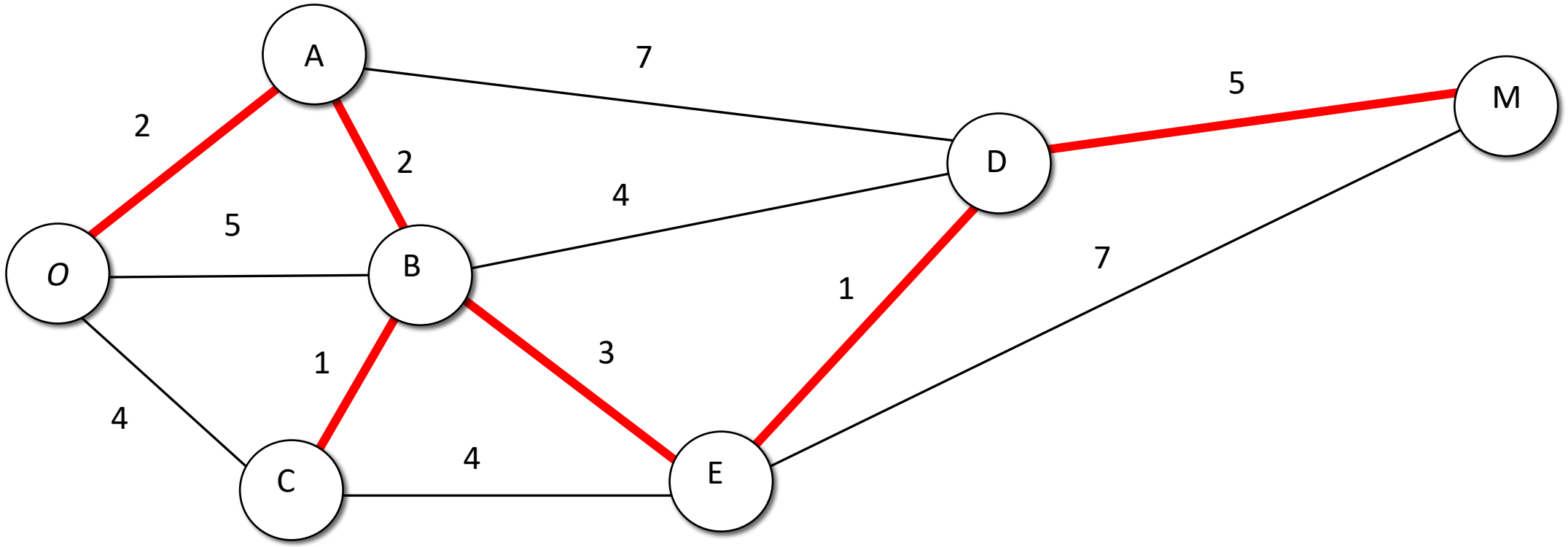
Minimum Spanning Tree – Start at O



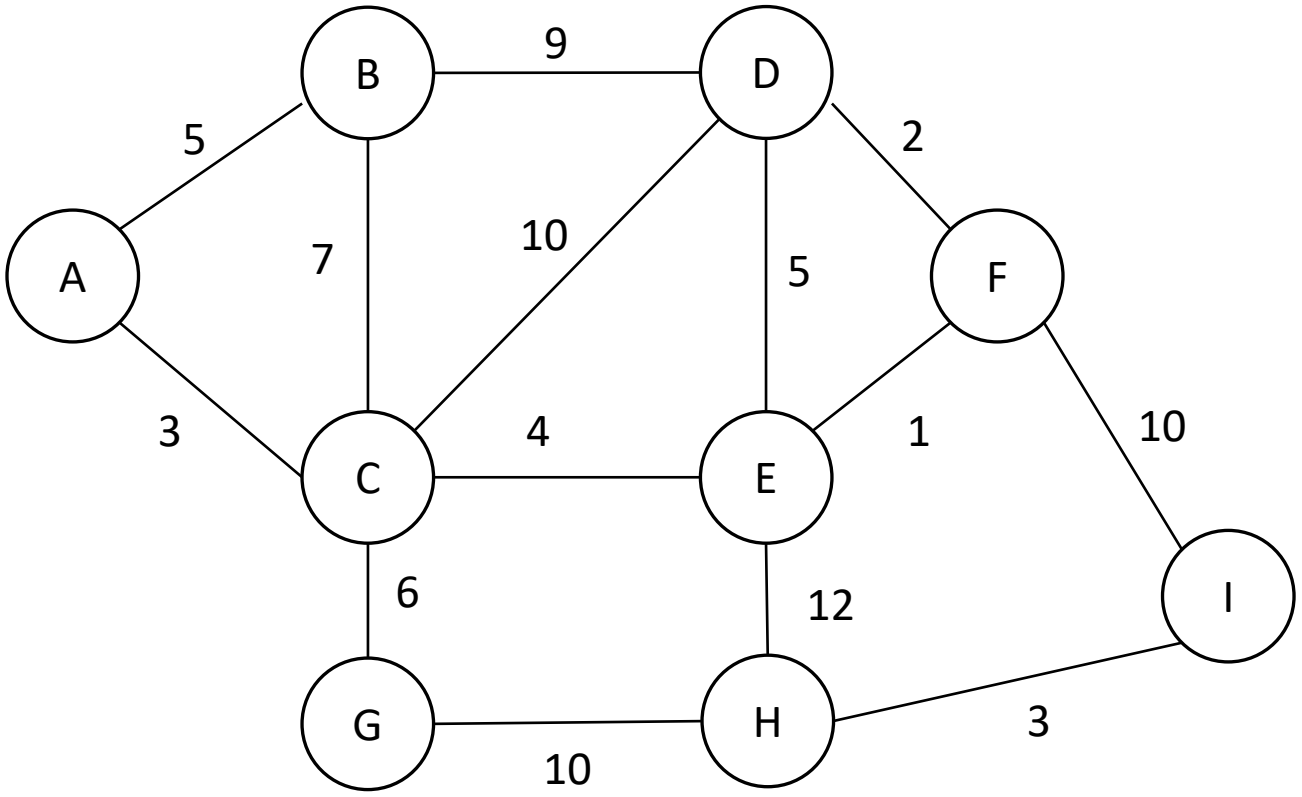
Minimum Spanning Tree – Start at O



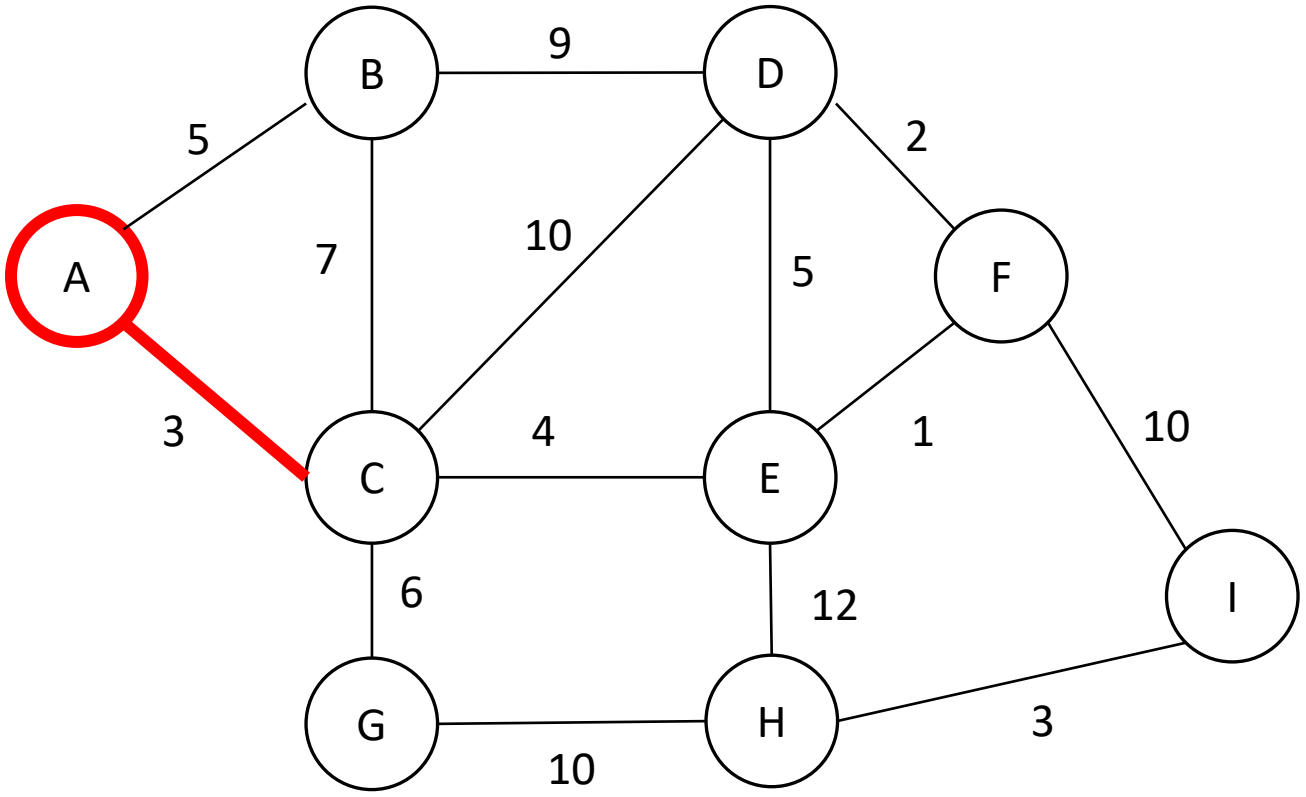
Minimum Spanning Tree – Start at O



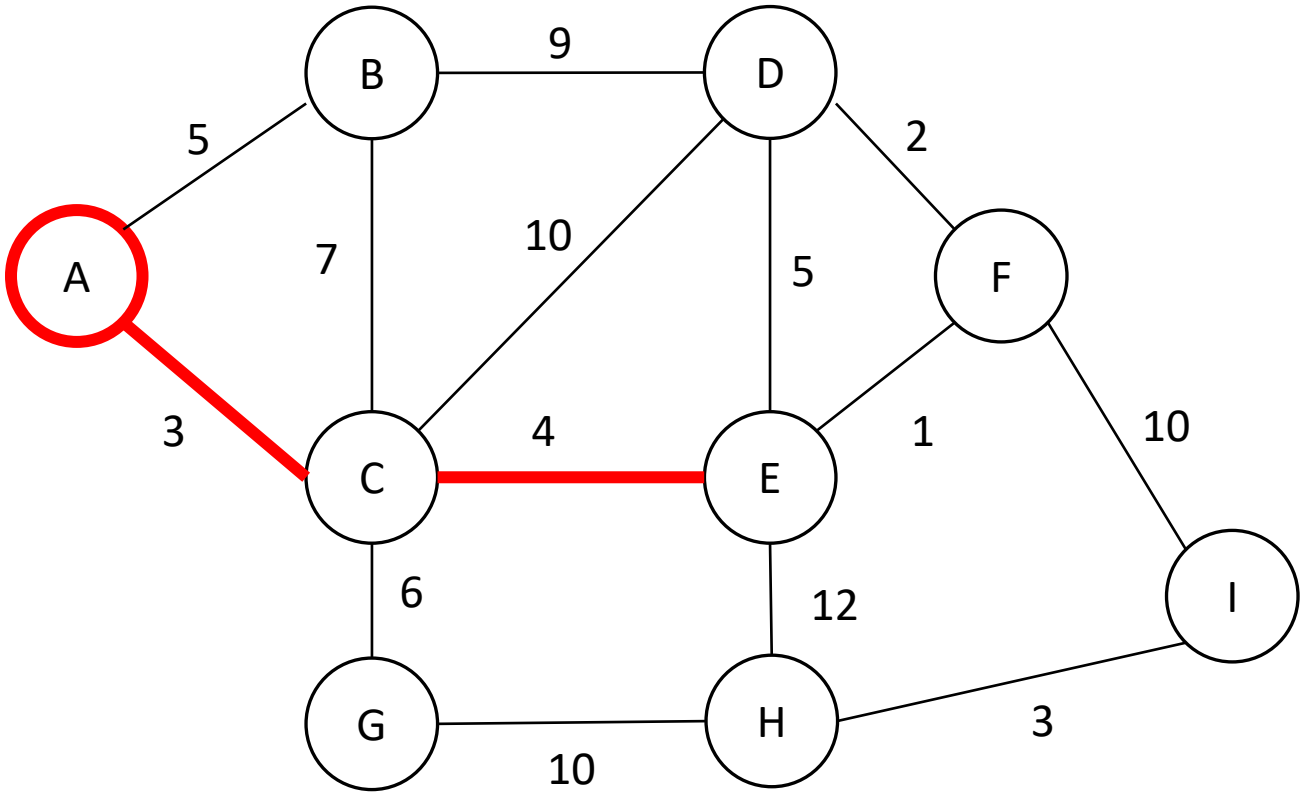
You Try It – Start at A



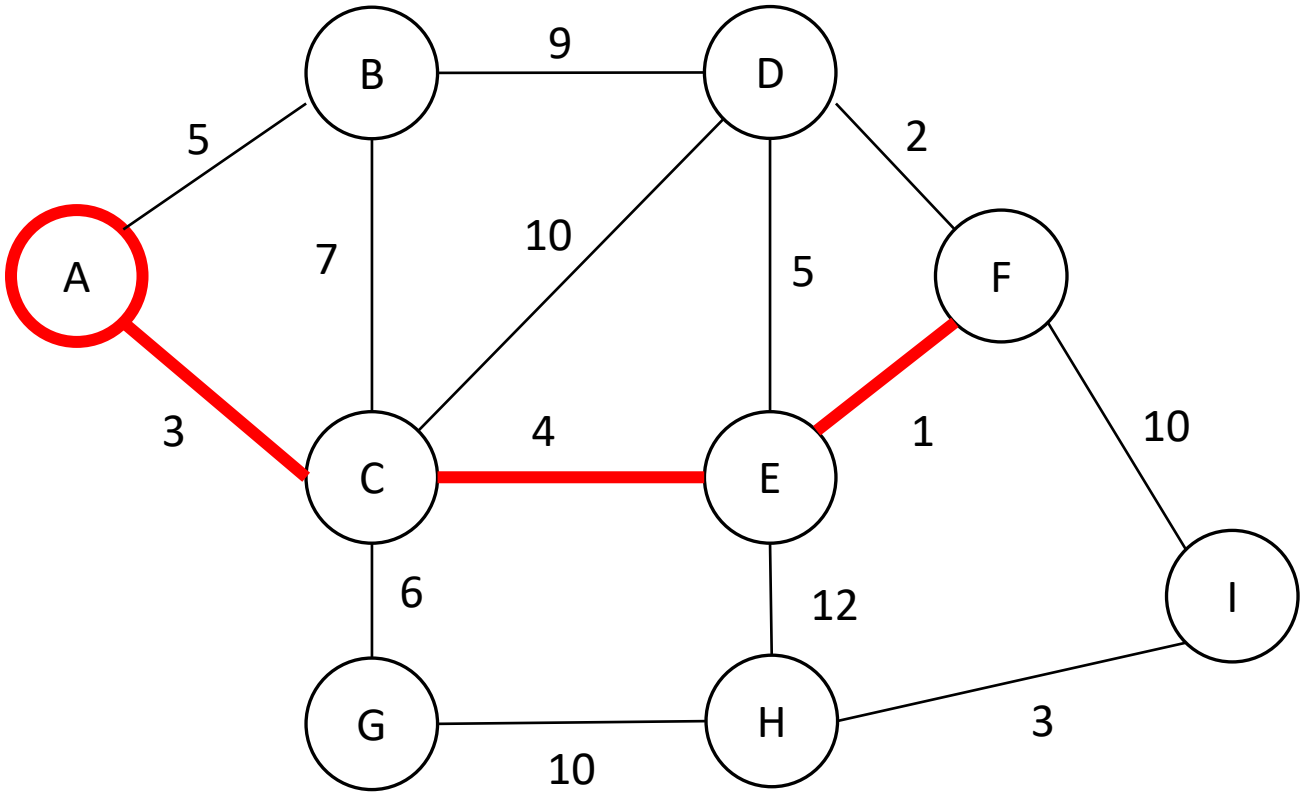
You Try it – Start at A



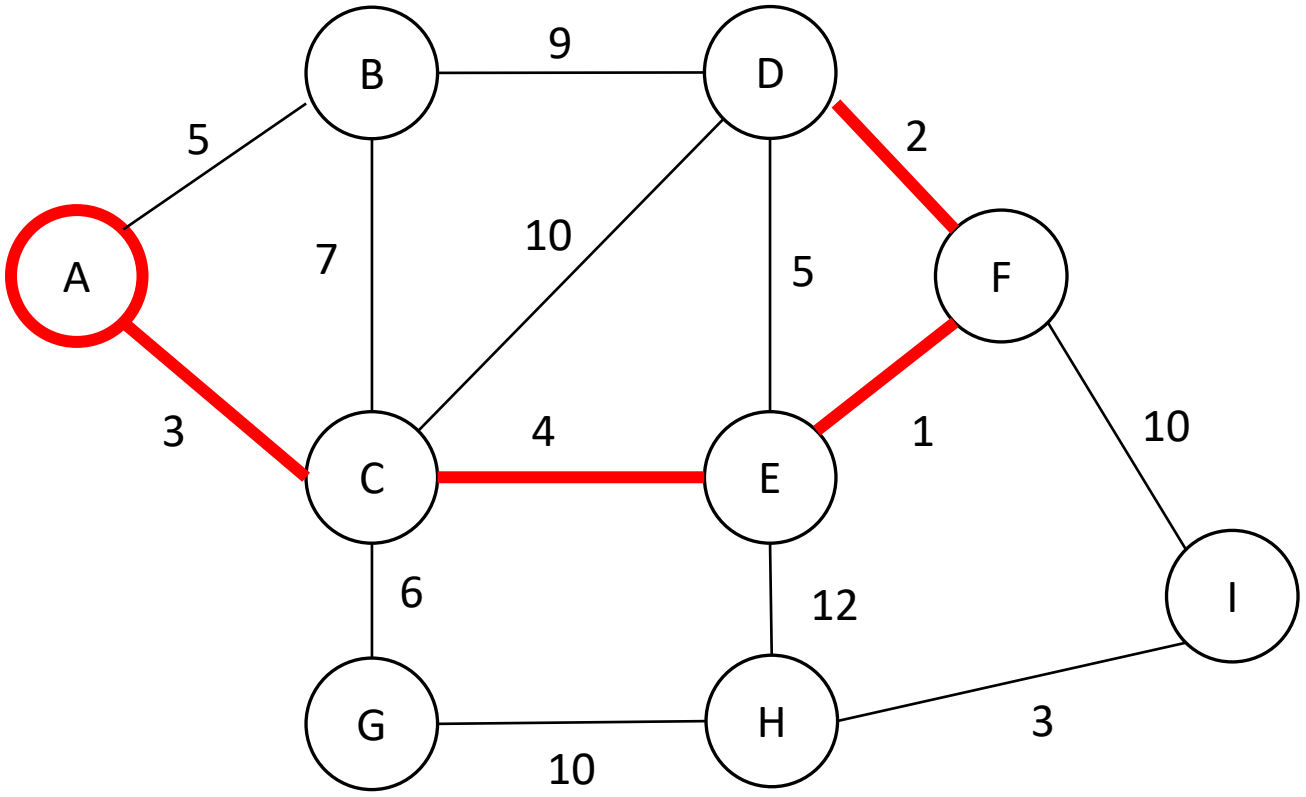
You Try it – Start at A



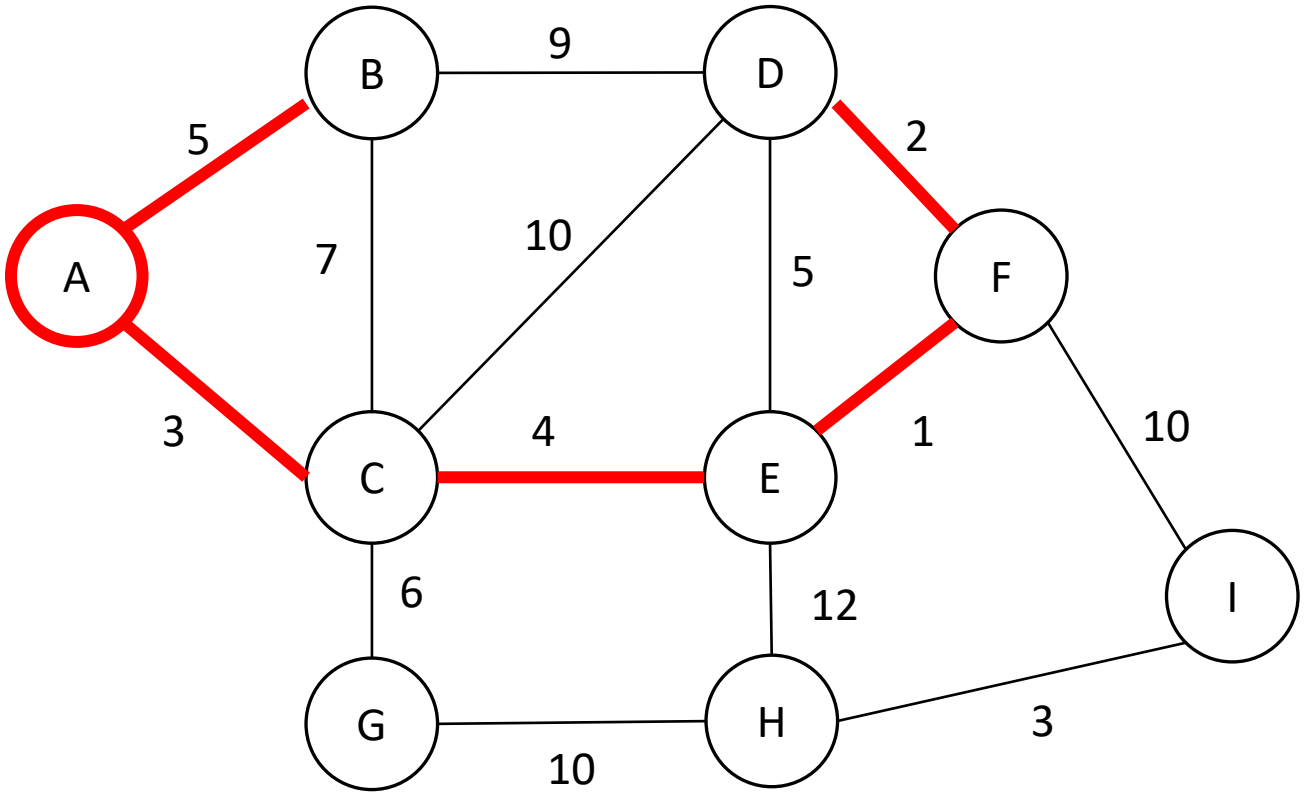
You Try it – Start at A



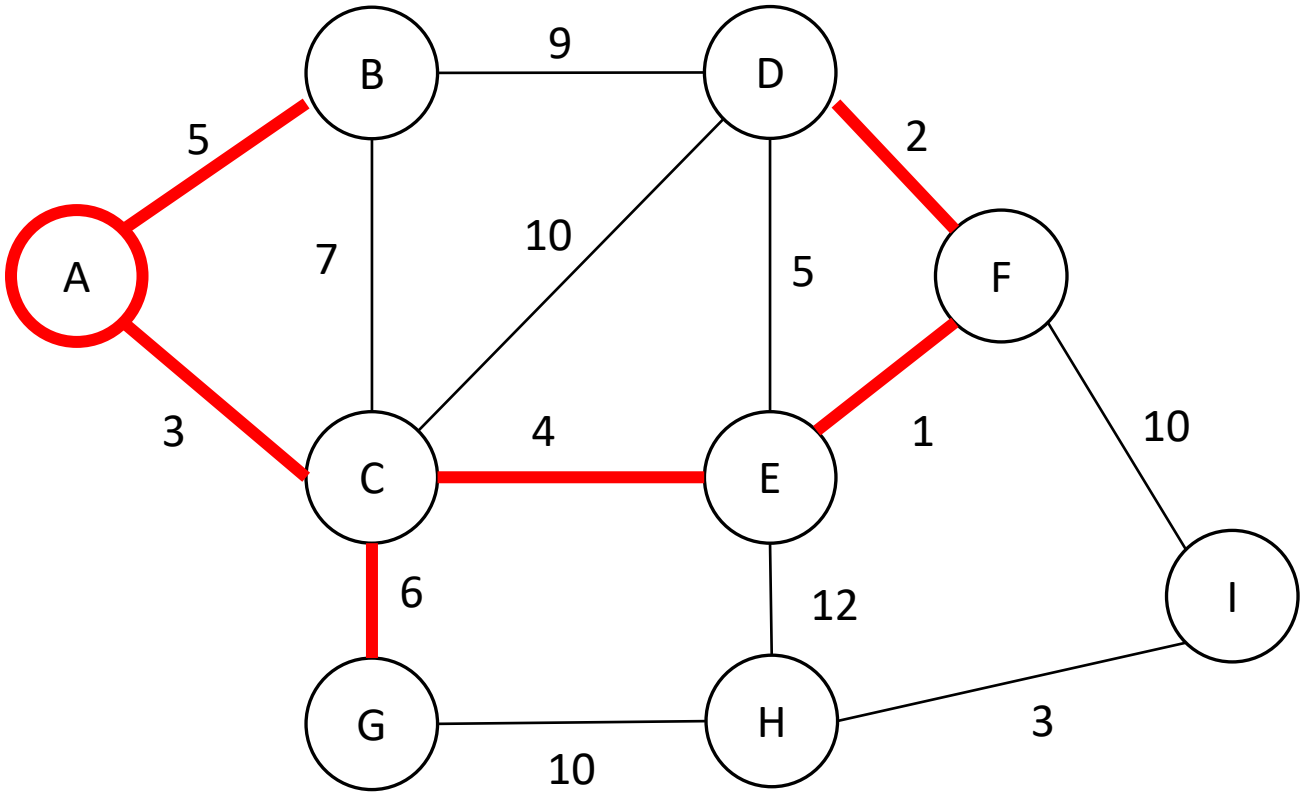
You Try it – Start at A



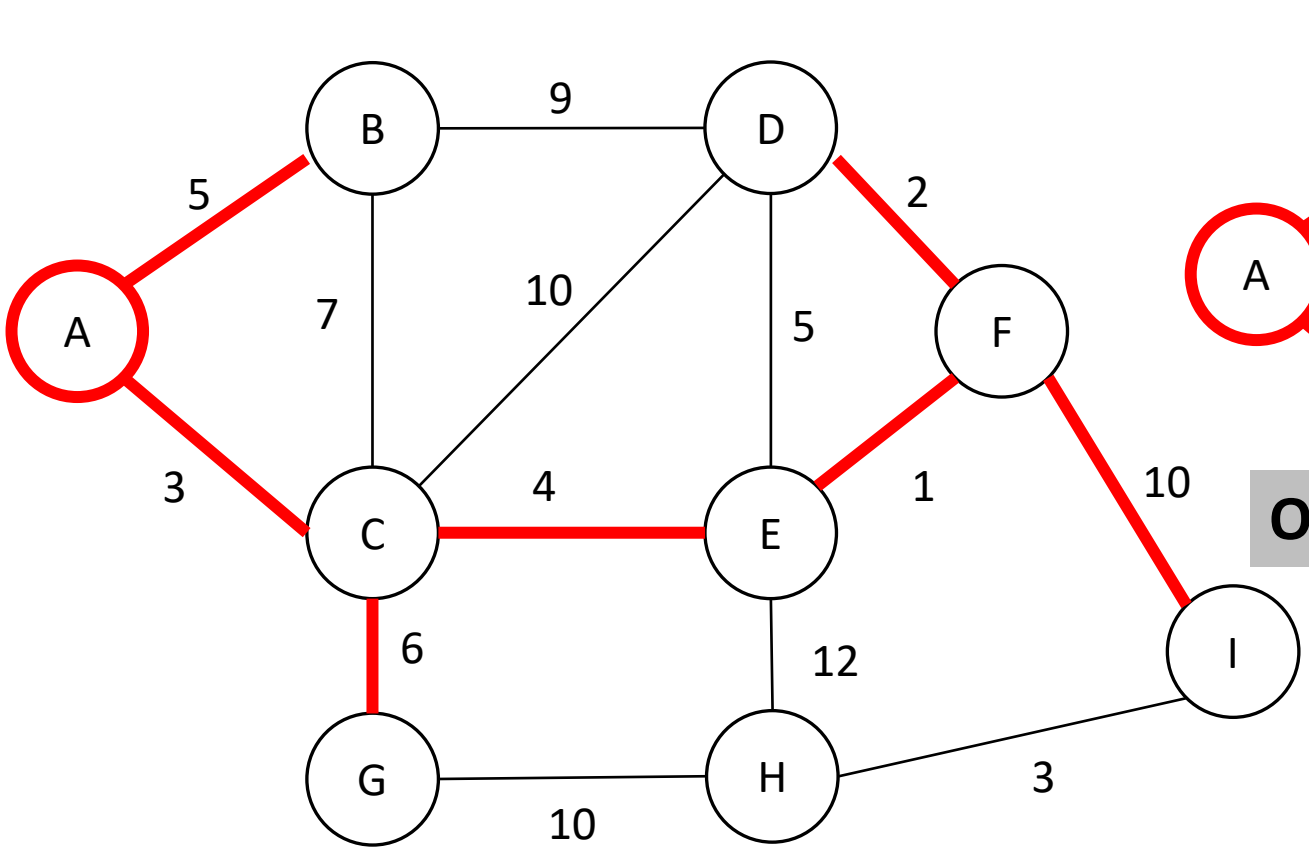
You Try it – Start at A



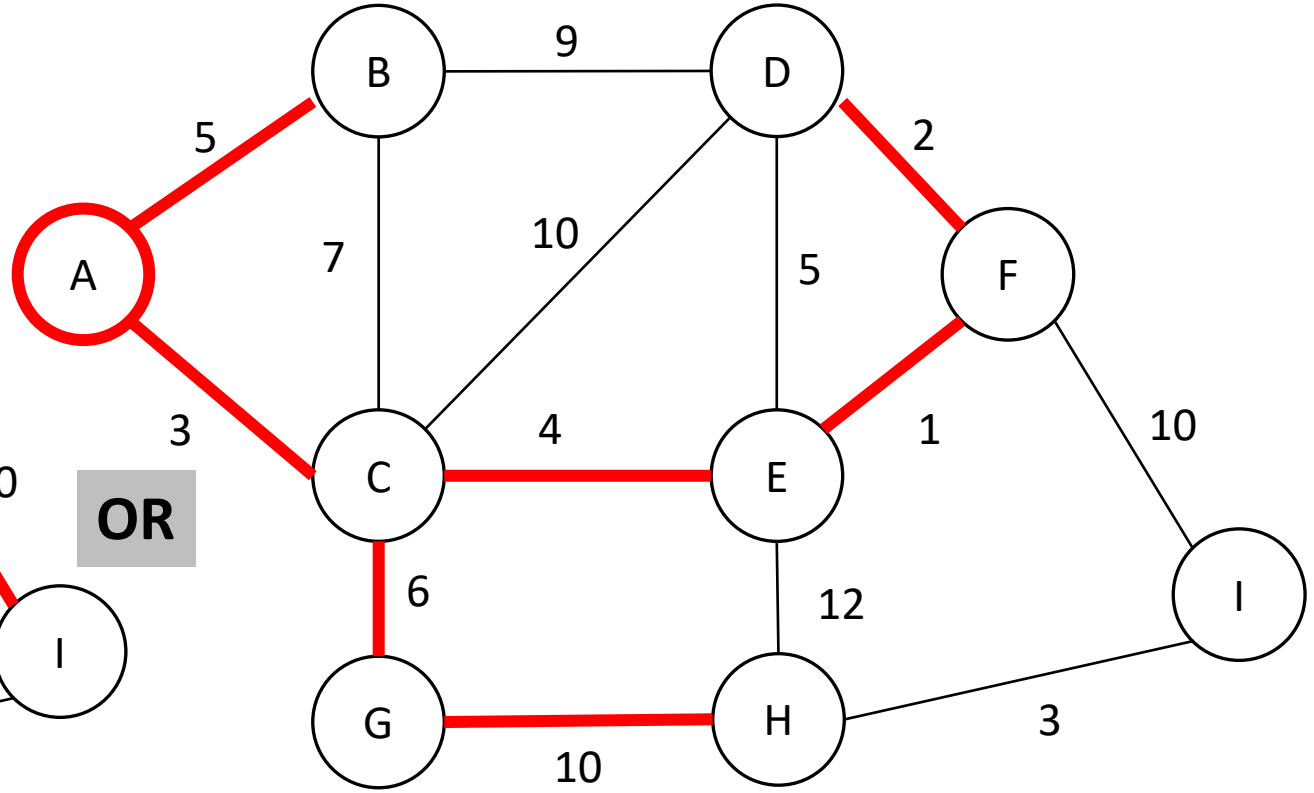
You Try it – Start at A



You Try it – Start at A



OR



You Try it – Start at A