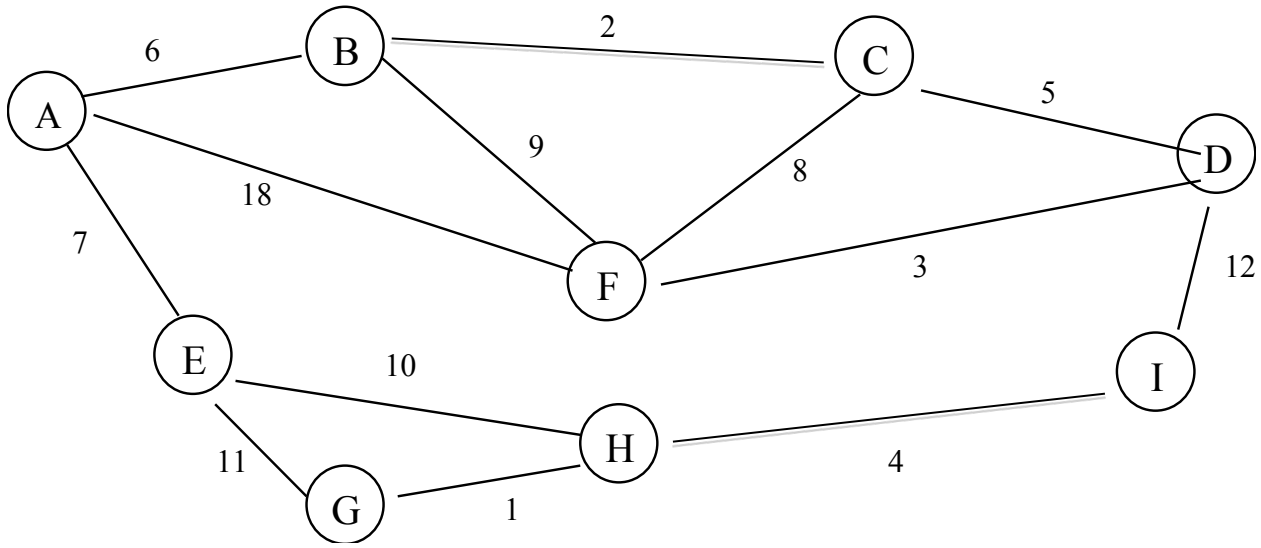
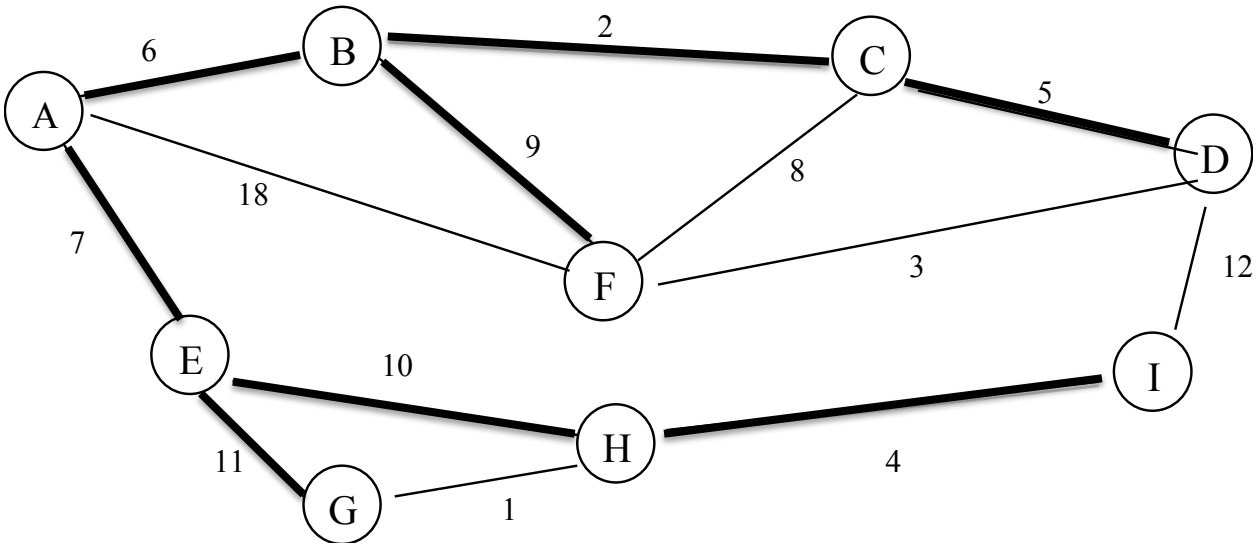
## CS 1501 Directed Weighted Graph and Shortest Path Tree

Original graph showing edge weights




Since the graph is a digraph, the actual representation for the above graph will include two edges for each link shown. For example, edge AB, 6 above is actually stored as edge AB, 6 in A's adjacency list and as edge BA, 6 in B's adjacency list. There is no requirement for this symmetry in a digraph. However, I have done it here so that this graph can be compared with the previous MST graph.

Shortest path spanning tree (with A as the source) edges shown in **BOLD**. At each iteration, **the next closest vertex to A** is chosen. Note that even if the edge weights are unique, the SP tree may not be unique. For example, there are two shortest paths to G (AEG, 18 and AEHG, 18). The algorithm will choose the first one it comes upon. Also note that the edges in the shortest path tree do not necessarily match the edges in the MST. Interestingly, in this case the smallest edge in the overall graph (GH, 1) is not even in the tree. A table with more trace information related to Dijkstra's SP algorithm is on the next page.



| Vertex | Order Chosen | Distance | Path |
|---|---|---|---|
| **A** | **0** | **0** | **-** |
| **B** | **1** | **6** | **AB** |
| **C** | **3** | **8** | **ABC** |
| **D** | **4** | **13** | **ABCD** |
| **E** | **2** | **7** | **AE** |
| **F** | **5** | **15** | **ABF** |
| **G** | **7** | **18** | **AEG** |
| **H** | **6** | **17** | **AEH** |
| **I** | **8** | **21** | **AEHI** |