

John Hodson
UFID 5244-0415
jhodson@ufl.edu

My code may be compiled and run under any environment in which the JDK and the javac compiler is available.

My implementation is structured as five files: linkstate.java, Graph.java, Node.java, Edge.java, and DistanceTableEntry.java.

linkstate is the file that was implemented (as requested by the project spec) as the top-level file. This top-level program may be invoked by compiling all of the source code (use **javac *.java** from the project directory) and then using **java linkstate network.txt**, where network.txt is the name of the text file representing the graph.

Graph is the top-level representation of the graph structure and is created by passing the constructor the name of the file. The file is parsed into the Graph structure line-by-line (edge-by-edge). The graph is represented as a map, where the key is the ID of a node and the value is the actual node.

The Node data structure is represented as a set of edges and an ID. An API is made available to add an outgoing edge and get all of the outgoing edges. I overrode the equals() and hashCode() functions so that I could use this data structure in other data structures such as sets and maps. I also overrode the compareTo() method so that two Nodes may be compared to each other. This was used when sorting the distance table for output to the console.

The Edge data structure is represented as a source Node, a destination Node, and a weight. I overrode the equals() and hashCode() functions so that I could use this data structure in other data structures such as sets and maps.

I also created a very small "wrapper" data structure, called DistanceTableEntry, used to pair together a Node as the parent (predecessor) along with its corresponding distance. This was useful when constructing the distance table.

I tested my program on the CISE storm server. The output of my program for the large example given in the program specification can be seen in the first screenshot below. The second screenshot is a smaller graph, which I chose in order to verify the behavior of the program.

