

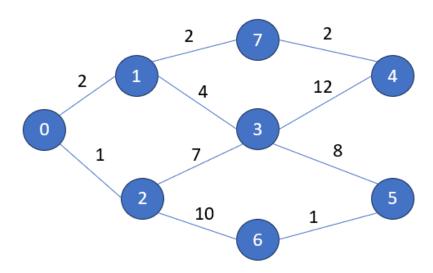
### Start Assignment

**Due** Nov 14 by 5pm **Points** 125 **Submitting** a file upload **File Types** java **Available** until Nov 14 at 10pm

A2 is in Three Parts - Part 1 is posted now, Parts 2 and 3 will be posted this weekend. Each part will be worth equal points and be of similar challenge.

#### Part 1 :: Dijkstra's Algorithm. (50 Points)

In this programming assignment, you are asked to implement the Dijkstra's algorithm in a Java program. The name of your submitted file should be **A2Part1.java**. Your Java program must be configured as a Java 11 Single Source Executable. This means that you should be able tot type java <A2Part1.java> and the Java compiler will run your program as a script. I will post a link to either a video or a web page explaining what is required. The format of the input network topologies is specified in an Nx3 table where N is the number of edges in the input topology. The values in the first and second columns in the table show the node IDs between edges while the values in the third column indicate the costs of edges. Furthermore, node IDs are always numbered between 0 and N-1 for a topology consisting of N nodes. The values in first column are always sorted in order from 0 to N-1. There is no duplicate edge in the table. In other words, if there is a row "1 2 8", there will not be a row "2 1 8". As example, the table below would represent the topology given beneath the table.



#### **Node1 Node2 Cost**

012

021

134

172

237

2610

3 4 12

358

4 7 10

561

# Input to your program:

- 1. a file consisting of a table with format describe above.
  - The name of the file must be specified as the first command line argument.
- 2. a source node ID.
  - The source node ID must be the second command line argument.

#### Output of your program:

Your program should print out a list of shortest paths from the source node to all other nodes and their associated costs with one destination path and cost per line. For example, the shortest path between node 0 and 7 in the Figure above is "0 1 7" and the associated cost is 4. You program should then print, on a newline, "0 1 7 4." Use only space separators.

The grader will run your program with different input files and source nodes to determine the correctness of your program.

# Part 2: Kruskal's Algorithm (50 Points)

In part 2 of this programming assignment, you are asked to implement Kruskal's algorithm in a Java program to give a minimum spanning tree of a given network. The name of your submitted file should be **A2Part2.java**. Your Java program must be configured as a Java 11 Single Source Executable. This means that you should be able tot type java **<A2Part2.java>** and the Java compiler will run your program as a script. I will post a link to either a video or a web page explaining what is required. The format of the input network topologies is specified in an Nx3 table where N is the number of edges in the input topology. Note this is exactly the same as part 1. The values in the first and second columns in the table show the node IDs between edges while the values in the third column indicate the costs of edges. Furthermore, node IDs are always numbered between 0 and N-1 for a topology consisting of N nodes. The values in first column are always sorted in order from 0 to N-1.

#### Node1 Node2 Cost

012

021

134

172

237

2610

3 4 12

358

4710

561

### Input to your program:

- 1. a file consisting of a table with format describe above.
  - The name of the file must be specified as the first command line argument.

# Output of your program:

Your program should print out a list of edges that describe the minimum spanning tree. Each line of output will contain the start node and the edge cost, taken from the input. The very last line of output will display a single number that represents the minimum cost of the spanning tree. For example, the edge 0,2 from above will be on the spanning tree so it will be included in the output as 0 2 1. This will be followed by the remaining edges of the minimum spanning three and the minimum cost of the tree on a line by itself.

The grader will run your program with different input files and source nodes to determine the correctness of your program.

### Part 3: Bellman Ford and Dynamic Programming (75 Points)

In part 3 of this programming assignment, you are asked to implement the Bellman Ford algorithm in a Java program to give the shortest paths to all nodes from a single source. The name of your submitted file should be **A2Part3.java**. Your Java program must be configured as a Java 11 Single Source Executable. This means that you should be able tot type java **<A2Part3java>** and the Java compiler will run your program as a script. I will post a link to either a video or a web page explaining what is required. The format of the input network topologies is specified in an Nx3 table where N is the number of edges in the input topology. The values in the first and second columns in the table show the node IDs between edges while the values in the third column indicate the costs of edges. Furthermore, node IDs are always numbered between 0 and N-1 for a topology consisting of N nodes.

#### Node1 Node2 Cost

012

021

134

172

237

2610

3 4 12

358

4710

561

## Input to your program:

- 1. a file consisting of a table with format describe above.
  - The name of the file must be specified as the first command line argument.
- 2. a source node ID.
  - The source node ID must be the second command line argument.

#### **Output of your program:**

Your program should print out a list of shortest paths consisting of node numbers separated by spaces that describe the shortest path. Each line of output will contain the start node and each node along the edge of the path followed by the cost of the path.

The grader will run your program with different input files and source nodes to determine the correctness of your program.

#### **Submission Instructions**

Submit three java files with names given exactly as above. Do not worry about Canvas appending revision numbers to the names, my script can correct for that. It cannot correct for arbitrary edits and so you will lose points if you don't follow the naming directions.