

- Home
- About Me
- My Publications
- Research
- Courses

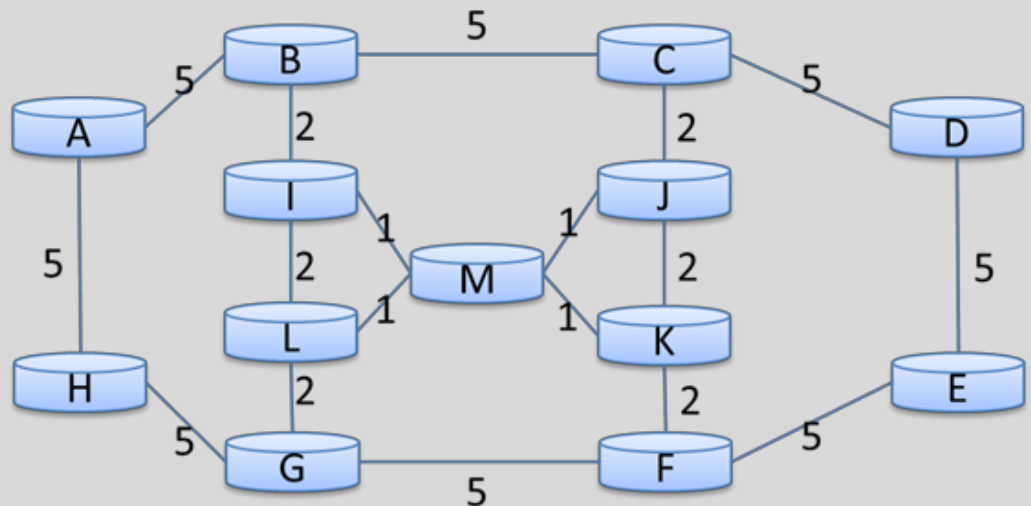
Networks: Implementing Dijkstra's Algorithm

Motivation

To cement the concepts surrounding Dijkstra's Algorithm and finding the shortest path through a network (graph).

Big Picture

You will implement a Dijkstra's algorithm and be able to determine the shortest path between any two nodes in a network. The network you will be dealing with can be visualized below.



You will be provided with a network definition file. That file will be a comma separated representation of the following matrix. Your program will read in this comma separated file and use it to build the above network. The program should take as arguments two letters. These will be the two nodes between which you wish to find the shortest path. The output will be a list of nodes that must be traversed (the first and last should be the two letters that were passed as arguments).

X	A	B	C	D	E	F	G	H	I	J	K	L	M
A	0	5	0	0	0	0	0	5	0	0	0	0	0
B	5	0	5	0	0	0	0	0	2	0	0	0	0
C	0	5	0	5	0	0	0	0	0	2	0	0	0
D	0	0	5	0	5	0	0	0	0	0	0	0	0
E	0	0	0	5	0	5	0	0	0	0	0	0	0
F	0	0	0	0	5	0	5	0	0	0	2	0	0
G	0	0	0	0	0	5	0	5	0	0	0	2	0

H	5	0	0	0	0	0	5	0	0	0	0	0	0
I	0	2	0	0	0	0	0	0	0	0	0	2	1
J	0	0	2	0	0	0	0	0	0	2	0	0	1
K	0	0	0	0	0	2	0	0	0	2	0	0	1
L	0	0	0	0	0	0	2	0	2	0	0	0	1
M	0	0	0	0	0	0	0	0	1	1	1	1	0

Here is a link to the above comma separated file ([network.csv](#)).

Specifics

Write a Java program that reads in network.csv and instantiates a network with the indicated costs.

Have it take two letters as arguments (source node followed by destination node).

Have it output to the screen a list of nodes that must be traversed in order to follow the lowest cost path from the source to destination node.

If you wish, you may make it a GUI program in which case it will need to have an interface that takes as input the source and destination nodes and somehow displays the path, but this is not necessary. A simple console program is sufficient.

When I ran my program...

I got the following output when I ran my implementation checking the distance between H and C

```
determining shortest path between H and C
H G L M J C
total distance: 11
```

When I ran A to E:

```
determining shortest path between A and E
A B I M K F E
total distance: 16
```

Implementation hints

I created a Node class and a GetShortestPath class (the GetShortestPath class had my main method). My node object had the following attributes:

- label (A, B, C, etc.)
- prevNode (reference to prev Node)
- minSoFar (integer)
- hashMap (hash of distances to all other nodes, use labels as keys)

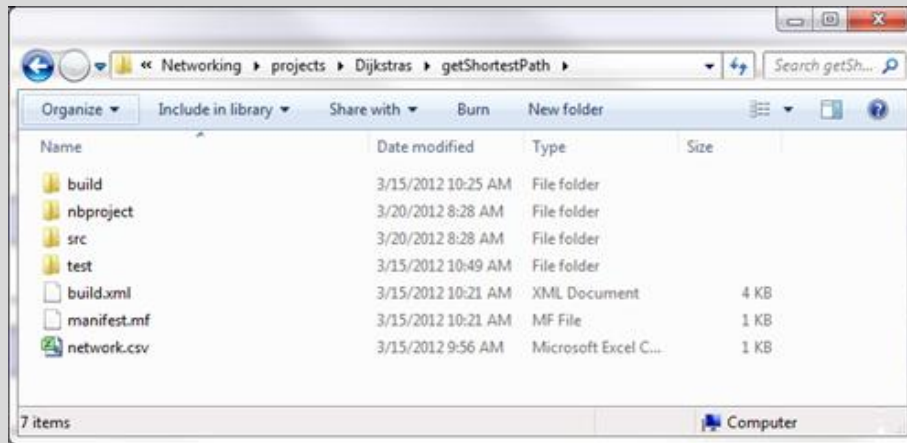
Within my GetShortestPath class I had the following methods:

- readInFile: generates in memory hash of hashes with the from label as the first key and the to node as the second key
- buildNetwork: instantiates each of the nodes and populates its hash of distances (using the above hash of hashes)
- determineShortestPath: Dijkstra's algorithm (has a completedNodes Set. Also uses a Stack in order to push labels of last node and all nodes that are encountered when follow prevNodes back to source node)
- getKnownMinNode: Dijkstra's helper, visit each node in completedNodes Set, find neighbors, calculate distance to that node (current node minSoFar+edge dist to that node), see if less than that node's minSoFar, if so, update, keep track of which adjacent node has minimal distance, and add that one to completedNodes.

Deliverables

You will turn in a Java project and a Readme document. The readme should include your name, the date, the name of the class and project. I used NetBeans and it generated a folder like the following. Note that I had to place the network.csv in directory so my readInFile function would work

NOTE that I had to place the network.csv in directory so my readMe function would work.



In the readme file include a description of how to run the program. If you use passed arguments indicate as much. If you did not use NetBeans (for instance, if you used Eclipse or a simple text editor) indicate where in the directory structure the .java files are so that I can compile and run them from the command line. Ensure you test this approach to ensure all imports can be found.

Provide a list of functions/methods that were written to support your project. Describe whether or not the program works as it should. If it does not, describe where you think the problem lies.

Grade Rubric

D work:

- Source code and a README document
- The README has the required information
 - A description of how to run the program and then how to run it including any required arguments, etc.
 - A list of functions/methods that were written to support your project.
 - Describe whether or not the program works as it should. If it does not, describe where you think the problem lies.

C work:

- All of the criteria of a D program must be met plus...
- The program performs all tasks described in the project correctly.
 - It finds the shortest path from source to destination adding nodes to an N' set (in my code, the completedNodes Set)
 - It calculates the distance correctly (the minSoFar value in the destination node at the completion of the run)

B work:

- All of the criteria of a C program must be met plus...
- The program outputs the correct path to the terminal screen (or GUI if desired)
- The program outputs the correct distance to the screen

A work:

- All of the criteria of a B program must be met plus...
- The code compiles with no errors or warnings.
- The code is well organized and easy to follow.
- The code is populated with comments. Every method is commented and virtually every line of code is accompanied with a comment, though possibly small chunks of code are commented instead.

Zip your files into a single file for submission (source, and readme file).

Submission Instructions

1. Go to [UM Online](#)
2. Click on the link "[Dijkstra's Algorithm project submission link. Due midnight Wednesday night March 28. Assignment](#)"
3. Type any comments you wish.
4. Click on the Browse button to navigate to the directory containing your files for this project.
5. Highlight files you wish to upload.
6. If you have additional files to submit, repeat.
7. When you have finished adding all the files, click on **Send for grading**.