

CSE 4344 Assignment 2 Spring 2020

Nihar Gupte

1001556441

For windows users:

To compile the program, use the following command in the local directory:

- `javac lab2.java`

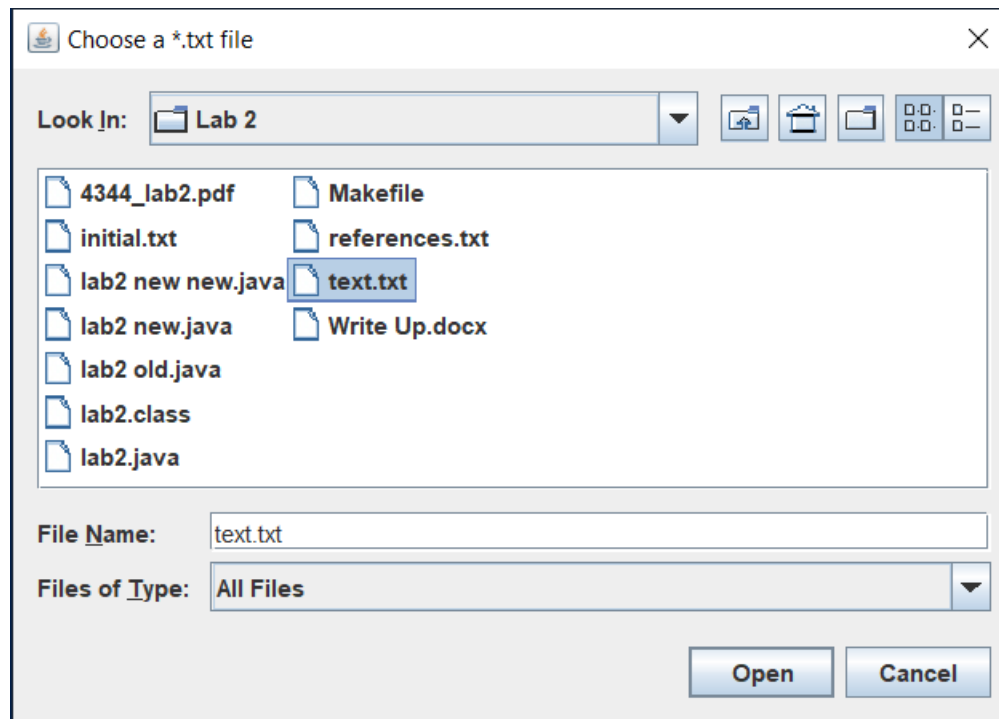
To execute the program, use the following command:

- `java lab2`

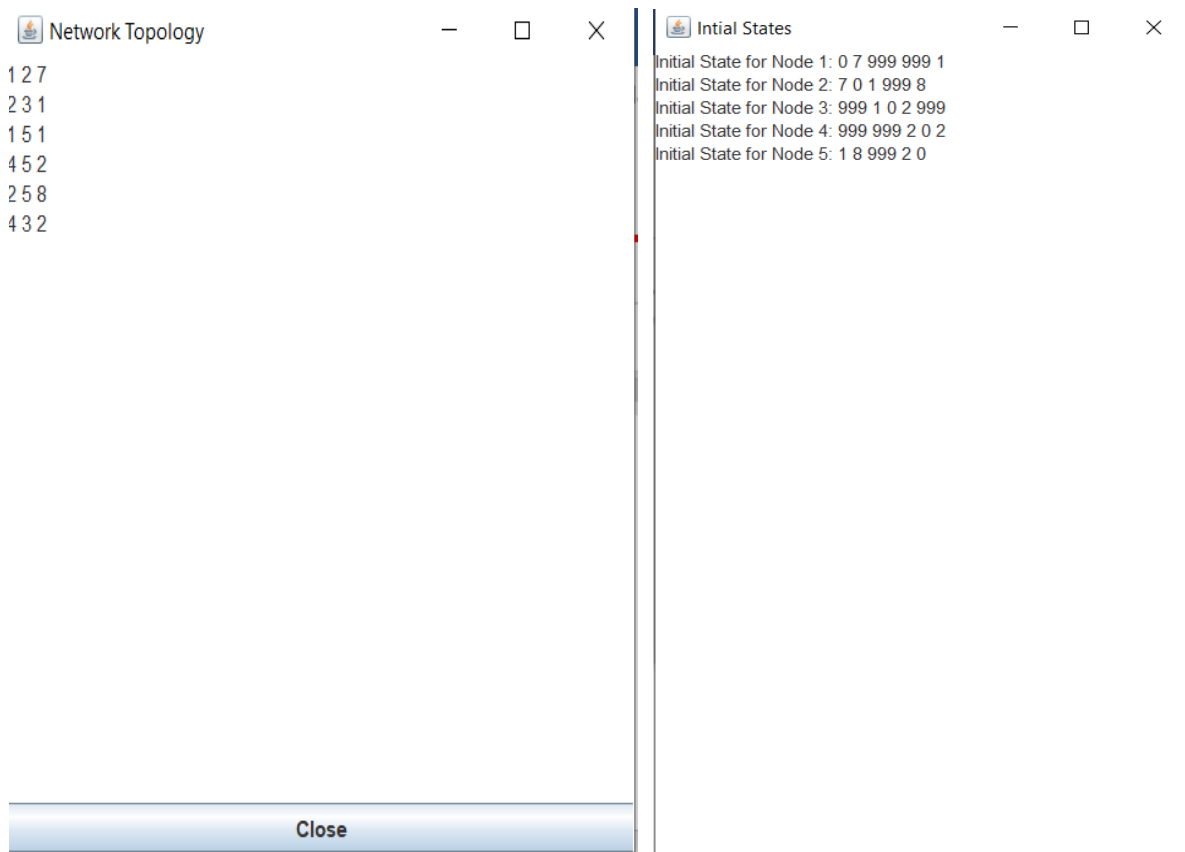
(A makefile is also included for linux based OS)

Steps:

1. Select a File from the File Chooser Menu. This should be the file containing the **network topology stored in *.txt format**. For the purpose of this documentation, **I have provided a sample “text.txt” file containing the network topology in the appropriate format.**




2. Once you have selected the file, two JAVA GUI windows should appear along with a dialog box. One of the JAVA GUI window will display the **network topology file** that you just selected, and the second JAVA GUI window will display the **Initial States for all Nodes in the graph**. I have set the **MAX_VALUE to 999**, so this is a limitation of the program as I am running under the assumption that the maximum path cost can only be 998. I am using **999 as a placeholder for infinity**. So, if Node 1 and 2 have 999 in their adjacency matrix / initial distance vector state table, that implies that 1 and 2 are not connected (not neighbors)



- Input ×

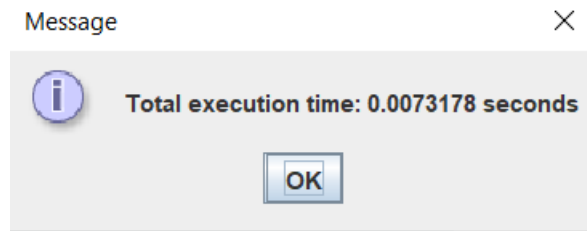
Input ✕

 **Type 1 for Single Step Mode and 2 for Non Stop Mode**

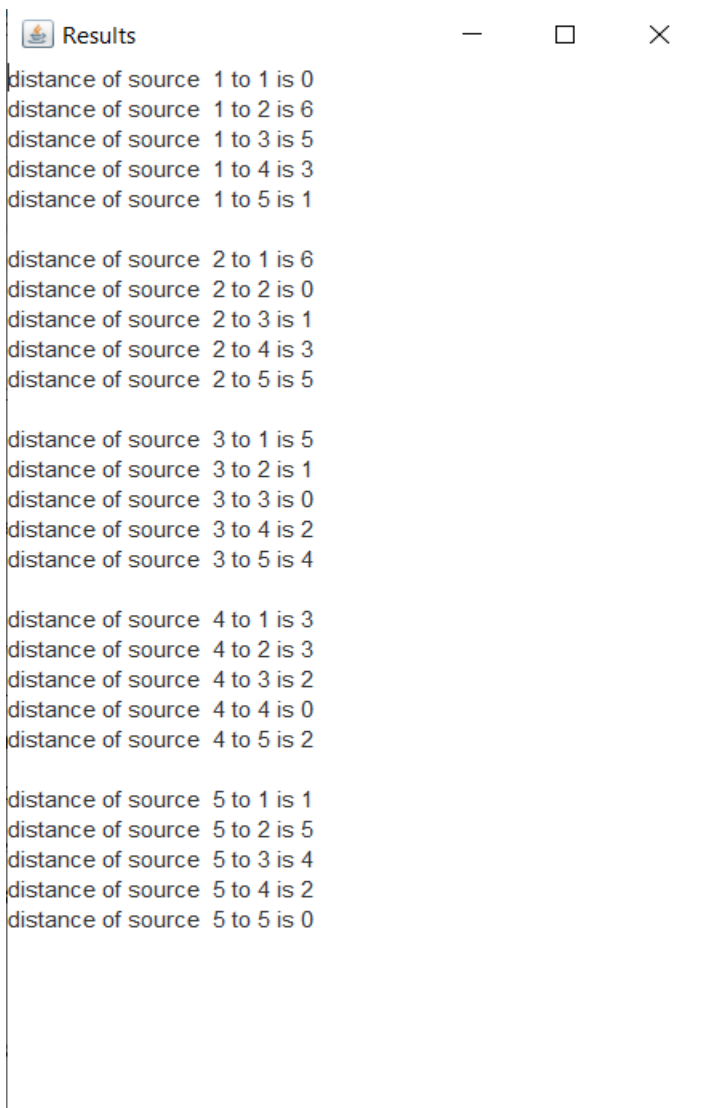
- Distance Vector Table for Node 3



- b. For Nonstop mode, you will get a dialog box showing you the **total time of execution in seconds** that was required to calculate the least cost path from all vertices to every other vertex in the graph.

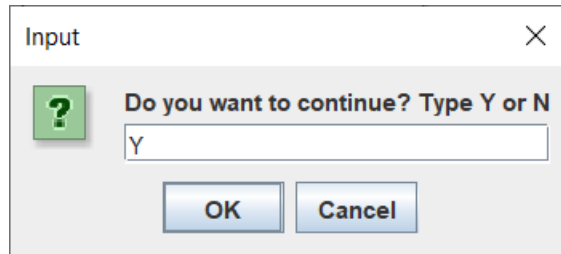


4. Finally, a last JAVA GUI window will pop up with the results displayed in it which shall **display the shortest path from every source to every vertex.**

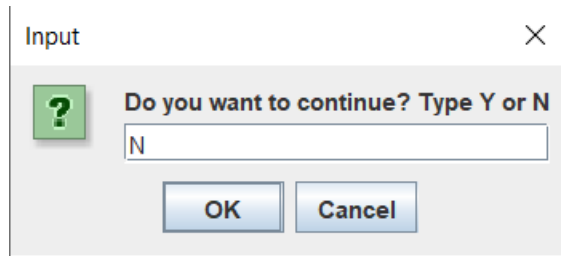


5. The above-mentioned information remains on the screen for 15 seconds and then another dialog box shows up asking the user if he wants to select another network topology and run the same program on it again.

a. If the user types 'Y', then repeat Step 1 to 5.



b. If the user types 'N', the program terminates itself in 2 minutes (under the assumption that the reader is able to understand the data in these 2 minutes)



References for code:

<https://stackoverflow.com/questions/16802147/java-i-want-to-read-a-file-name-from-command-line-then-use-a-bufferedReader-to>

http://www.java2s.com/Tutorials/Java/Swing_How_to/JFileChooser/Display_the_Contents_of_a_text_file_in_a_JTextArea.htm

<https://www.sanfoundry.com/java-program-implement-bellmanford-algorithm/>

<https://www.youtube.com/watch?v=9VrtranTJnc>

<https://gist.github.com/jintujacob/1024755>

<https://www.tutorialspoint.com/how-to-close-jframe-on-the-click-of-a-button-in-java>

https://www.youtube.com/watch?v=2_sx1OPwPOs

<https://stackoverflow.com/questions/3382954/measure-execution-time-for-a-java-method>