Indian Institute of Technology Roorkee
Department of Computer Science and Engineering

CSN-261: Data Structures Laboratory
(Autumn 2019-2020)
Lab Assignment-8 (L8)

Name :- Harshit Verma

Branch :- CSE

Enroll no:- 18114029

Sub Batch :- O2

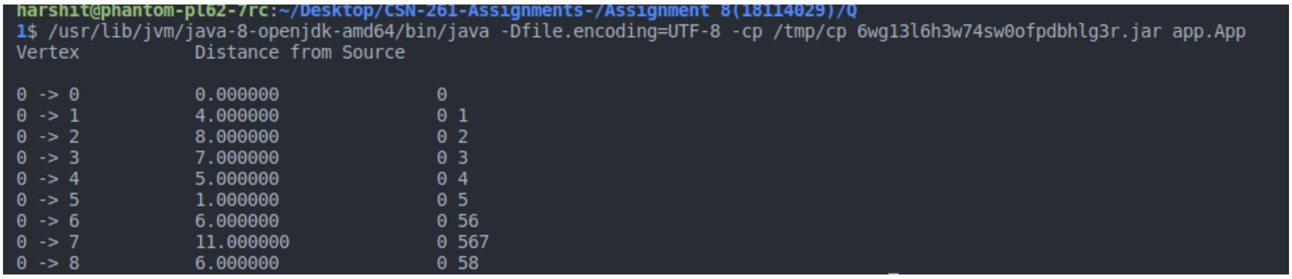
Problem Statement 1:

Implement Dijkstra’s algorithm in Java to find all shortest paths between all pair of vertices in a weighted graph. Modify this algorithm to find all shortest paths between two nodes, if more than one occurs. Following this, compute betweenness centrality measure of each node. Betweenness Centrality of a node/vertex, w is given as , where, ഉട is the number of all shortest paths between u and v; and ഉടሺ.𝑤ሻ is the number of all shortest paths between u and v through w. (https://en.wikipedia.org/wiki/Betweenness_centrality) Data structure that may be used: List, Set, Map, etc. Input: A GML (Graph Modeling Language) file as a graph input. Output: Betweenness Centrality of each node. Note: Use JGraphT class in java (https://jgrapht.org) for this problem**.**

Data-Structure: Hashmap, Priority queue/Min heap.

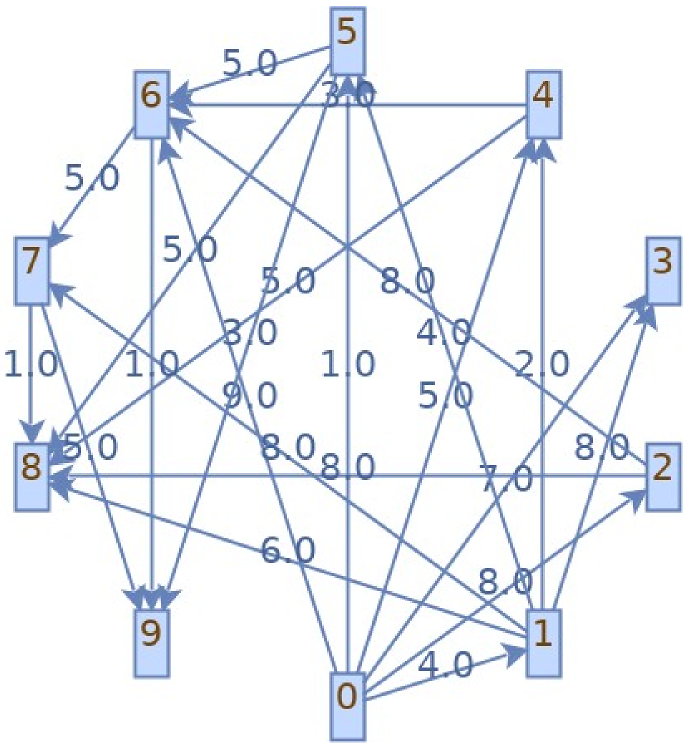
Algorithm: Dijkstra.

Screenshots:



```
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 8(18114029)/Q
1$ /usr/lib/jvm/java-8-openjdk-amd64/bin/java -Dfile.encoding=UTF-8 -cp /tmp/cp 6wg13l6h3w74sw0ofpdbhlg3r.jar app.App
Vertex           Distance from Source

0 -> 0           0.000000                0
0 -> 1           4.000000                0 1
0 -> 2           8.000000                0 2
0 -> 3           7.000000                0 3
0 -> 4           5.000000                0 4
0 -> 5           1.000000                0 5
0 -> 6           6.000000                0 56
0 -> 7           11.000000               0 567
0 -> 8           6.000000                0 58
```



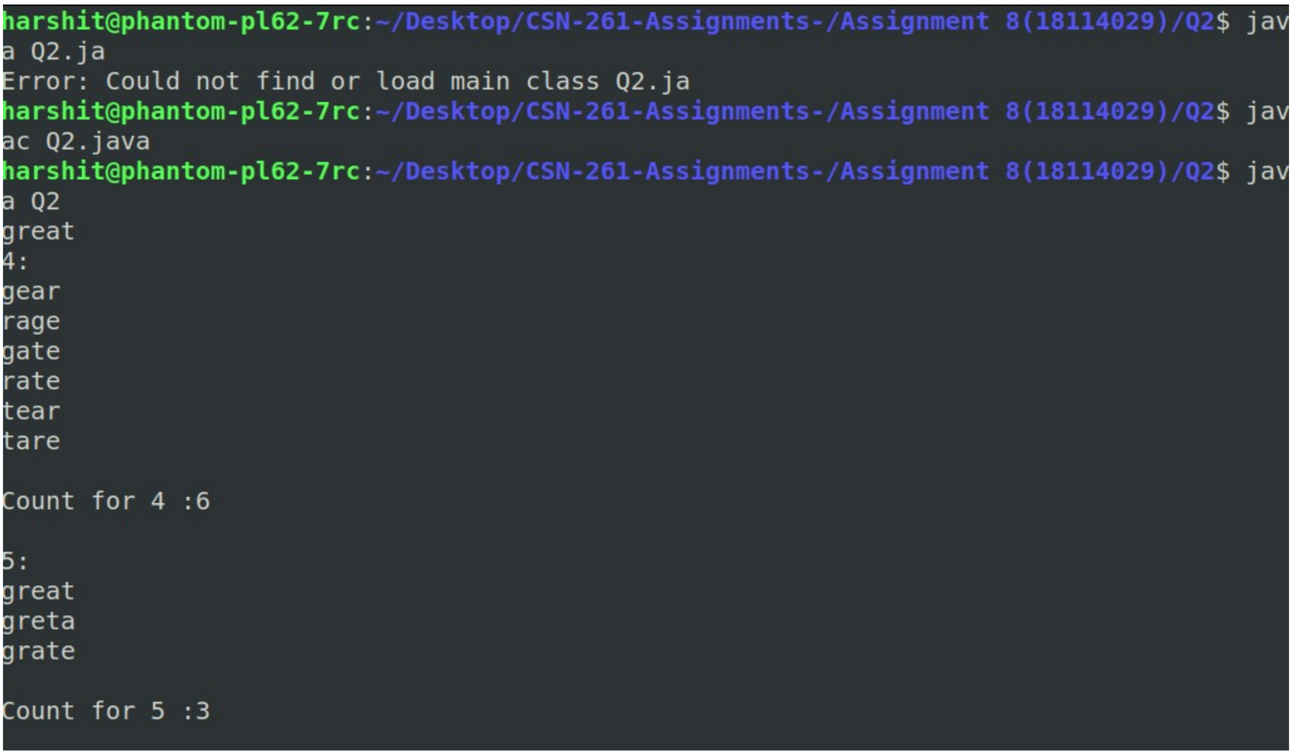
Problem Statement 2:

Create a project/program in Java called Unscramble Word. Given a string of 'N' characters print all the words present in a dictionary of length 'M' such that 3 < M <= N. Use dictionary present in Linux @ /usr/share/dict/words. Implement this code in java and the student may use inbuilt data structures such as Maps, Sets, etc. (For fast execution, use of Trie is suggested). Input: A String Output: All unscrambled words of given string present in the dictionary categorized by length of word. Also print the total number of words of each length.

Data-Structure: Trie.

Algorithm: Divide and Conquer.

Screenshots:



```
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 8(18114029)/Q2$ jav
a Q2.ja
Error: Could not find or load main class Q2.ja
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 8(18114029)/Q2$ jav
ac Q2.java
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 8(18114029)/Q2$ jav
a Q2
great
4:
gear
rage
gate
rate
tear
tare

Count for 4 :6

5:
great
greta
grate

Count for 5 :3
```