ASSIGNMENT 8

Data Structures Laboratory

Rishi Chordia

18118052

BTech CSE

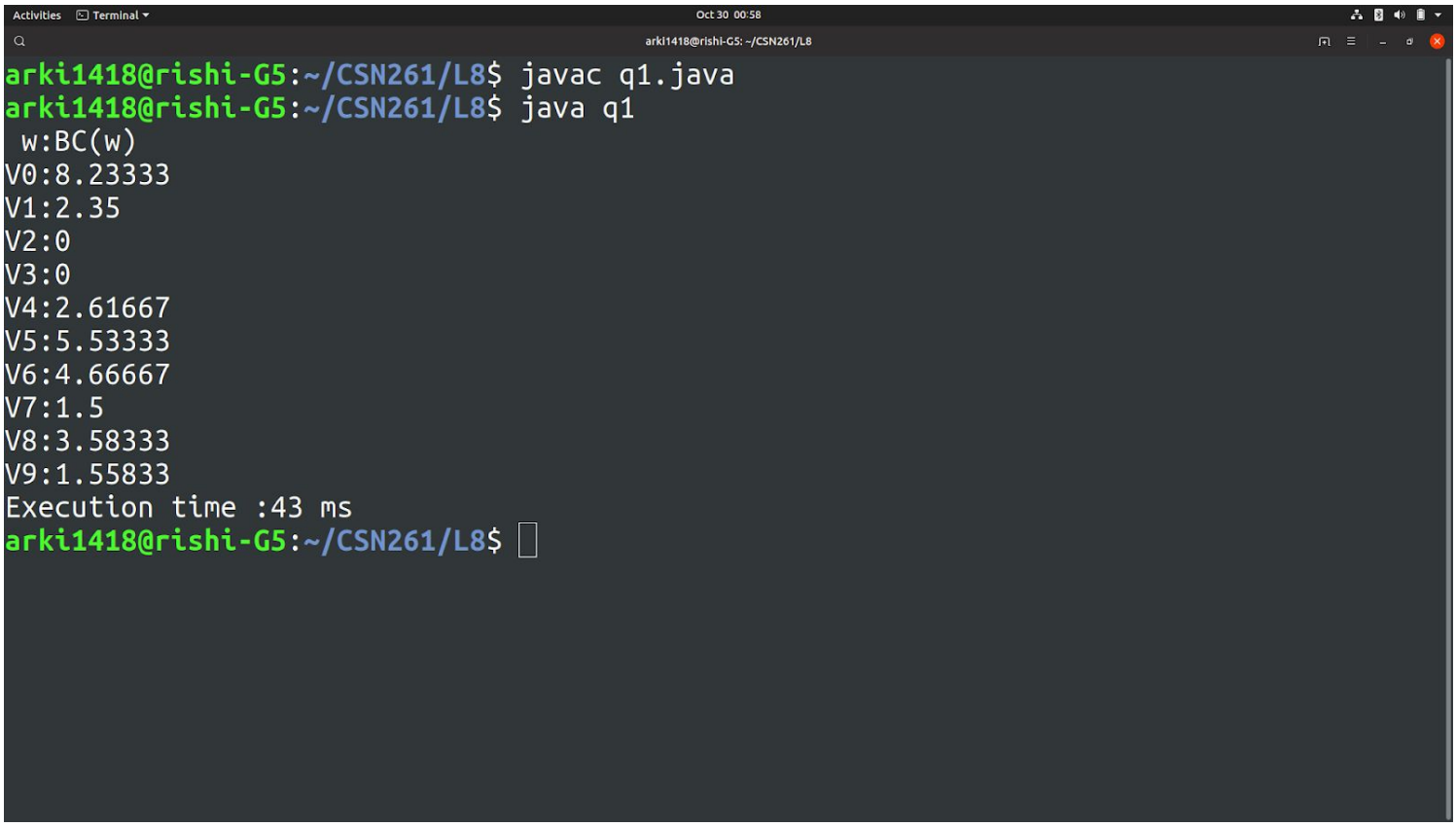
Problem Statement 1:

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 which is the number of all shortest paths between u and v summed over all u and v divided by the number of all shortest paths between u and v through w.

Algorithms and Implementation:

- **A slightly modified Dijkstra's Algorithm is implemented**
- **JGraphT graph libraries are used to represent the graph**
- **Sets and Maps are used in implementation of Dijkstra on graph**
- **Betweeness Centrality of each node is calculated in a recursive manner to find all shortest paths.**

Snapshots and Computation Time



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.

Algorithms and Implementation:

- **Maps are used to keep a track of the words in the dictionary**
- **Techniques of recursion have been used to generate all the permutation of a string of a given length.**

Snapshots and Computation Time