

# *Lab Assignment - 8*

Name: ANMOL KHEMUKA

Enrollment No: 18114007

Batch: O1

## **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

([https://en.wikipedia.org/wiki/Betweenness\\_centrality](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 STRUCTURES USED:**

- Graphs
- List
- Map
- Arrays
- ArrayList

## **ALGORITHMS USED:**

- Dijkstra's Algorithm with little modification

```
$ java -cp "jgrapht-core-1.3.0.jar:jgrapht-io-1.3.0.jar:antlr4-runtime-4.7.1.jar:commons-lang3-3.8.1.jar:commons-text-1.5.jar:jgrapht-ext-1.3.0.jar:jgraphx-3.4.1.3.jar:jheaps-0.9.jar" q1.java L8_P1_sample.gml
The Betweenness Centrality of node 0 : 7.8333
The Betweenness Centrality of node 1 : 1.3333
The Betweenness Centrality of node 2 : 6.75
The Betweenness Centrality of node 3 : 2.6667
The Betweenness Centrality of node 4 : 0.0
The Betweenness Centrality of node 5 : 9.0
The Betweenness Centrality of node 6 : 0.0
The Betweenness Centrality of node 7 : 1.75
The Betweenness Centrality of node 8 : 7.0
The Betweenness Centrality of node 9 : 6.25
```

## **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  $3 < M \leq N$ .

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 STRUCTURES USED:**

- ArrayList
- Map
- HashSet
- Arrays
- Trie

## **ALGORITHMS USED:**

- Recursive Algorithms to find Unscrambled Words

```
$ java Unscrambler dictionary.txt

Please enter word to be Unscrambled : great
Length: 4 geat, grat, trag, agre, rate, tera, tare, rage, gaet, geta, ga
te, gare, ager, gear, Count: 14
Length: 5 retag, targe, grate, great, gater, Count: 5
```