

**Azim Javedali Saiyed (800889017)**

# Project 2: Dijkstra's Algorithm

---

## Compiling & Executing the Java code:

Compile the java code with the following command:

***javac graph.java***

Execute the program with the following code:

***java graph network.txt***

I have successfully compiled the code and executed in the following environment:

OS: Windows 8.1

Compiler: javac 1.8.0\_40

## Program Design & Breakdown of File:

There is only one Java file named "graph.java" which does all the five tasks as required by the project.

I have implemented Dijkstra's algorithm using MinHeap class which calculates the shortest distance between any two vertices of the graph. Also I have implemented the Reachable Vertices algorithm without using the shortest path algorithm from the program. I have developed a recursive algorithm which calculates the reachable vertices in the graph in  $O(n \log n)$  time.

I have Used the following Classes in the program:

- **Graph** - Class which reads the file which is passed to it and stores the graph and performs all the operations on the graph.
- **GraphException** - Handles any RuntimeException thrown by the Graph.
- **Vertex** - Class which handles all the information related to the vertex.
- **Edge** - Class which stores the information related to the Edge.
- **Path** - Class which stores the name of a Path.
- **Pair** - Class which stores the String Pairs.
- **Minheap** - Class which implements the minheap for the Dijkstra's Algorithm.

## Compiler Used:

The compiler that I have used to compile the java program is "GCJ, the GNU compiler for java" and "javac" compiler.

### Summary of Program Sustainability:

The program works perfectly for the given inputs and I have also tried the algorithms for various other graphs and queries and it works for them as well.

### Data Structure Design:

I have used the following Data Structures for various purposes in the program:

- HashMap
- LinkedList
- ArrayList
- Arrays
- Pair (User Defined)
- Path (User Defined)
- TreeSet
- String
- Integer
- Boolean
- Vertex (User Defined)
- Edge (User Defined)