CSCI 3005 – Programming Assignment 3 – Spring 2016

Your task is to use the breadth-first search algorithm studied in class to implement a solution to the problem of finding paths in an unweighted, undirected graph. In developing your solution, you are to use the Graph.java class provided by the instructor.

Data for your program will be obtained from text files in which each line contains the names of two vertices corresponding to an edge in the graph (for instance, see contiguous-usa.dat for edges corresponding to adjacent states in the USA).

Your solution is to be implemented as a class named `PathFinder` containing the following public methods:

> `PathFinder(String filename)`: a constructor to read in the data from a text file which contains a series of lines. Each line has the names of two graph vertices separated by a single space.
>
> `int getPathLength(String source, String destination)`: returns the smallest number edges to be traveled in the path from source to destination. Throws an exception if any of the vertices are invalid. Returns -1 if no path exists.
>
> `String getPath(String source, String destination)`: returns the vertices in the shortest path from source to destination, using the format below. It should return "NONE" if there is no path.
>
> `[source --> vertex --> vertex --> destination]`
>
> `Set<String> reachable(String source, int distance)`: returns a set of strings containing vertices reachable by traveling no more than 'distance' edges from the source.

The PathFinderTest.java program and sample text files are available for partial testing. Submit your PathFinder.java (and any other .java files developed as part of your solution) as a single .zip file to Mimir for testing. Your solution .zip file should include neither Graph.java nor any data files nor any .class files).

Sample method calls using the Graph class:

```
Graph g = new Graph();
g.addVertex(25);
g.addVertex(10);
g.addVertex(50);
g.addEdge(25, 50);
g.addEdge(25, 10);
System.out.println(g);
```