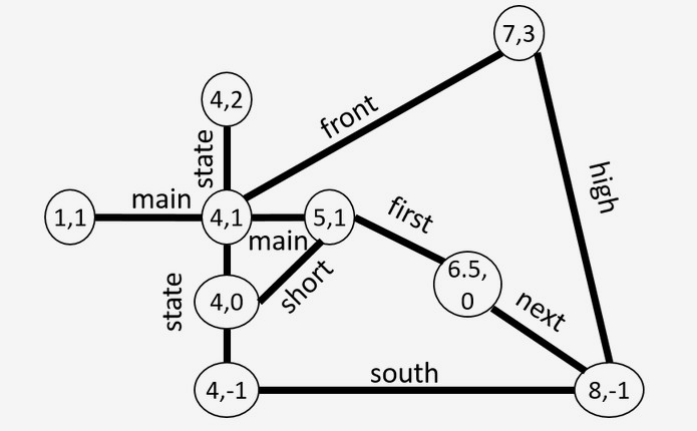
Computer Science	Graphs – BFS-Part 1	Lab Value - 100

Lab Goal : This lab was designed to teach you how to implement a graph with an adjacency list and matrix.

Lab Description : This lab is composed of two parts. In part one you'll design and implement a set of classes that will represent a graph structure with street data. Moreover you'll implement breadth first search and verify on the graph below. The shortest route(in steps) form (1,1) to (8,-1) is thru (4,1) and (7,3). You may add/build any classes that you deem necessary. I started an inner node and edge class that you may use. In part two you'll implement dijkstra's algorimth and add a more efficient way of searching. Do not modify any of the other classes.

The test data is represented pictorially below:



Files Needed ::
`mapGraph.java`

Sample Output :

```
Making a new map...DONE.
Loading the map...DONE.
Num nodes: 9
Num edges: 22
[Lat: 1.0, Lon: 1.0, Lat: 4.0, Lon: 1.0, Lat: 7.0, Lon: 3.0, Lat: 8.0, Lon: -1.0]

// and afterwards run SearchGrader

Score: 1.0
Feedback: All tests passed. Great job!

Straight line (0->1->2->3->...)
** Test #1: Testing vertex count...PASSED.
** Test #2: Testing edge count...PASSED.
** Test #3: Testing BFS...PASSED.

Same as above (searching from 6 to 0)
** Test #4: Testing vertex count...PASSED.
** Test #5: Testing edge count...PASSED.
** Test #6: Testing BFS...PASSED.

Square graph - Each edge has 2 nodes
** Test #7: Testing vertex count...PASSED.
** Test #8: Testing edge count...PASSED.
** Test #9: Testing BFS...PASSED.

UCSD MAP: Intersections around UCSD
** Test #10: Testing vertex count...PASSED.
```

```
** Test #11: Testing edge count...PASSED.
** Test #12: Testing BFS...PASSED.
```