**GIT Department of Computer Engineering**

**CSE 222/505 – Spring 2020**

**Homework #8 Report**

**Türker Tercan**

**171044032**

**Question 3:**

**Problem Solution:**

* We need to convert a given maze to a weighted not directed graph. The graph is given as a txt file. Where a 0 represents an open square and a 1 represents a closed one. and should find the shortest path from upper-left corner to lower-right corner.
* Firstly, I used an ArrayList of strings to store all strings in txt file.
* To initialize a graph, firstly, I need to count all the vertices and those are junction points in the txt file.
* Junction point is the breakpoint that the user can change its direction. And these are the junction point’s rules. This patterns makes their midpoint’s as junction
  + 101 111 101 101 101

000 Junction 000 Junction 000 Junction 100 Junction 001 Junction

101 101 111 101 101

* + 111 101

000 No junction 101 No junction

111 101

* I represent as junctions as two dimensional int array. When it found a junction in example column = a, row = b, junctions[b][a] = junction count and increment junction count as well.
* Then, we have how many junctions there are and their locations.
* Each vertices can be edged with another 4 vertices at most. Which means if in v vertex there can be 4 \* v edges. Which means if square of v divided by 2 is larger than 4 multiplied by v, graph is dense. Otherwise, the graph is sparse.
* I checked the graph is spare or not, then, according to it, I initialized my graph as adjacency matrix if the graph is dense, otherwise I initialized it as adjacency list.
* Insert the edges between connected junction points and their weight’s will be index or column size between them.
* I used dijkstra’s algorithm to find shortest way between first and last vertices.
* Then print the result.

**Test Cases:**

Test Subject: Given txt file converted to weighted map and successfully found shortest path

Test Number : T1

Pass/Fail: Passed

Running And Results:

Test T1:

Test Data:

Graph.txt:

011111111111111111111111

000000000000000000000001

011111111111111011111101

011111100000001011111101

011111101111111011000001

000000000000000011011011

110111101101111011011011

110111101101111011011011

110111101101000011011011

110111101101111111011011

110111101100000000011011

110000001101111111111011

111101111100000000001011

111101111111111111101000

111100000000000000001110

111111111111111111111110

SolveTheMaze(new Scanner(new File("Graph.txt")));

Expected:

Maze is solved

From upper-left corner to lower right corner minimum distance is: 40.0

0, 0 --> 1, 0 --> 1, 15 --> 1, 22 --> 4, 22 --> 4, 21 --> 13, 21 --> 13, 23 --> 15, 23 -->

Pass/Fail: Passed