# CSCE 221 Final Project Cover Page

First Name Suqian Last Name Wang UIN 825009505

User Name wangsuqian123 E-mail address wangsuqian123@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more on Aggie Honor System Office website: `http://aggiehonor.tamu.edu/`

| Type of sources | Lecture Sides | website | |
|---|---|---|---|
| People | Teresa Leyk | | |
| Web pages (provide URL) | | http://www.geeksforgeeks.org/ depth-first-traversal-for-a-graph/ | |
| Printed material | | | |
| Other Sources | | | |

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work. *On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.*

Your Name Suqian Wang Date August 5, 2017

# CSCE 221 Final Project
# Summer 2017

August 6, 2017

## 1 Data Structures

The data structure that I used to represent the graph is an adjacent list. The adjacent list is implemented using a vector of vectors.

## 2 Graph Operations

make_graph(): create a graph based on the maze.

search_path(): search paths from entry to exit in the graph.

show_path(): display each path from entry to exit in the text format, provide the length of the each path, and identify the shortest path.

show_graph(): print out the adjacent list for a given graph.

show_path_vertices(): print out all vertices for all entry-exit paths.

find_shortest_path(): find the shortest path of all path found from entry to exit.

## 3 Algorithm Description and running time analysis

I created a room class to store information of each room, including room number and door indentifer.

I created another class called map to store the map information including number of row column, the number of room, the entry id, the exit id, a vector path to store the paths found from entry to exit, a vector graph store the each room and its adjacent room, a vector store the room number of current path, and a vector of room type that store every room and their information.

The algorithm can perform several operation:

Assume there are N paths from entry to exit.

make_graph(): $O(n^2)$ For each room, add it's adjacent room to a vector. To traverse all the room, there will need $n^2$ operations.

search_path(): DFS time complexity is O(V+E), here $V = n^2$, and $E \leq 4n^2$. Therefore, this function's time complexity is $O(n^2)$.

show_path(): $O(Nn^2)$ For each path, mark "o" or "x" to every room need $n^2$ operations.

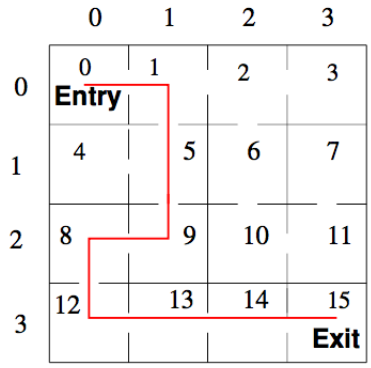
show_graph(): $O(n^2)$ The graph vector has n elements, every element is a vector which has at most 4 elements(4 doors), to show the graph, we have at most $4n^2$ operations.

show_path_vertices(): $O(Nn^2)$ There will be at most $n^2$ vertices on each path.

find_shortest_path(): $O(N)$ Compare the length of each path there will be N-1 comparision at most.

# 4 Test case and test result

**Test case 1:**




Input file:

16

0 1 0 0

0 1 1 1

0 1 1 1

0 0 0 1

0 1 0 0

1 0 1 1

1 0 1 0

0 0 1 0

0 1 1 0

1 0 0 1

1 1 0 0

1 0 0 1

1 1 0 0

0 1 0 1

0 1 0 1

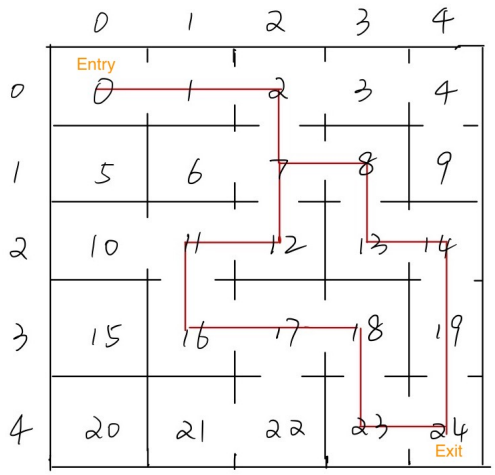
0 0 0 1

Test result:

SuqianWang — ssh -Y wangsuqian123@unix.cse.tamu.edu — 76×60

```
please enter the name of the input file: map04.txt
------------
You can do one of the following options, please enter the option number:
<1> print the adjacency list for a given graph.
<2> find the length of all entry-exit paths, and the shortest path
<3> print out all vertices for all entry-exit paths

enter your choice: 1
0->1
1->2->5->0
2->3->6->1
3->2
4->5
5->1->9->4
6->2->10
7->11
8->9->12
9->5->8
10->6->11
11->7->10
12->8->13
13->14->12
14->15->13
15->14
------------
You can do one of the following options, please enter the option number:
<1> print the adjacency list for a given graph.
<2> find the length of all entry-exit paths, and the shortest path
<3> print out all vertices for all entry-exit paths

enter your choice: 2
->display path 0
o o x x

x o x x

o o x x

o o o o

The length of the path above is 8

the shortest one is path 0, the length of which is 8
------------
You can do one of the following options, please enter the option number:
<1> print the adjacency list for a given graph.
<2> find the length of all entry-exit paths, and the shortest path
<3> print out all vertices for all entry-exit paths

enter your choice: 3
the vertices of the 0 path is:
0 1 5 9 8 12 13 14 15
------------
You can do one of the following options, please enter the option number:
<1> print the adjacency list for a given graph.
<2> find the length of all entry-exit paths, and the shortest path
<3> print out all vertices for all entry-exit paths

enter your choice: 4
Error: undefined option
```

**Test case 2:**




Input file:

25

0 1 0 0 0

1 0 1 0 1

1 1 0 1 0

1 0 0 1 1

0 0 0 0 0

1 0 0 1 1

1 1 0 0 1

1 1 0 0 0

0 1 0 0 0

1 1 1 1 0

0 1 1 1 0

0 0 0 1 1

0 0 0 0 1

1 0 0 0 1

1 1 0 0 1

1 1 0 1 0

0 0 0 0 0

1 0 0 1 0

0 1 1 1 0

0 1 0 0 1

Test result:

SuqianWang — ssh -Y wangsuqian123@unix.cse.tamu.edu — 79×58

```
[wangsuqian123]@linux2 ~/Final_Project> (01:59:28 08/06/17)
[:: g++ -std=c++11 *.cpp -o Main

[wangsuqian123]@linux2 ~/Final_Project> (01:59:36 08/06/17)
[:: ./Main
[please enter the name of the input file: map05.txt
------------
You can do one of the following options, please enter the option number:
<1> print the adjacency list for a given graph.
<2> find the length of all entry-exit paths, and the shortest path
<3> print out all vertices for all entry-exit paths

[enter your choice: 1
0->1
1->2->0
2->3->7->1
3->4->2
4->9->3
5
6->7
7->2->8->12->6
8->13->7
9->4
10->11
11->12->16->10
12->7->11
13->8->14
14->19->13
15
16->11->17
17->18->22->16
18->23->17
19->14->24
20
21->22
22->17->21
23->18->24
24->19->23
------------
You can do one of the following options, please enter the option number:
<1> print the adjacency list for a given graph.
<2> find the length of all entry-exit paths, and the shortest path
<3> print out all vertices for all entry-exit paths

[enter your choice: 2
->display path 0
o o o x x

x x o o x

x x x o o

x x x x o

x x x x o

The length of the path above is 8
```

SuqianWang — ssh -Y wangsuqian123@unix.cse.tamu.edu — 79×40

```
->display path 1
o o o x x

x x o x x

x o o x x

x o o o x

x x x o o

The length of the path above is 10

the shortest one is path 0, the length of which is 8
------------
You can do one of the following options, please enter the option number:
<1> print the adjacency list for a given graph.
<2> find the length of all entry-exit paths, and the shortest path
<3> print out all vertices for all entry-exit paths

enter your choice: 3
the vertices of the 0 path is:
0 1 2 7 8 13 14 19 24
the vertices of the 1 path is:
0 1 2 7 12 11 16 17 18 23 24
------------
You can do one of the following options, please enter the option number:
<1> print the adjacency list for a given graph.
<2> find the length of all entry-exit paths, and the shortest path
<3> print out all vertices for all entry-exit paths

enter your choice: 4
Error: undefined option
------------
You can do one of the following options, please enter the option number:
<1> print the adjacency list for a given graph.
<2> find the length of all entry-exit paths, and the shortest path
<3> print out all vertices for all entry-exit paths

enter your choice:
```