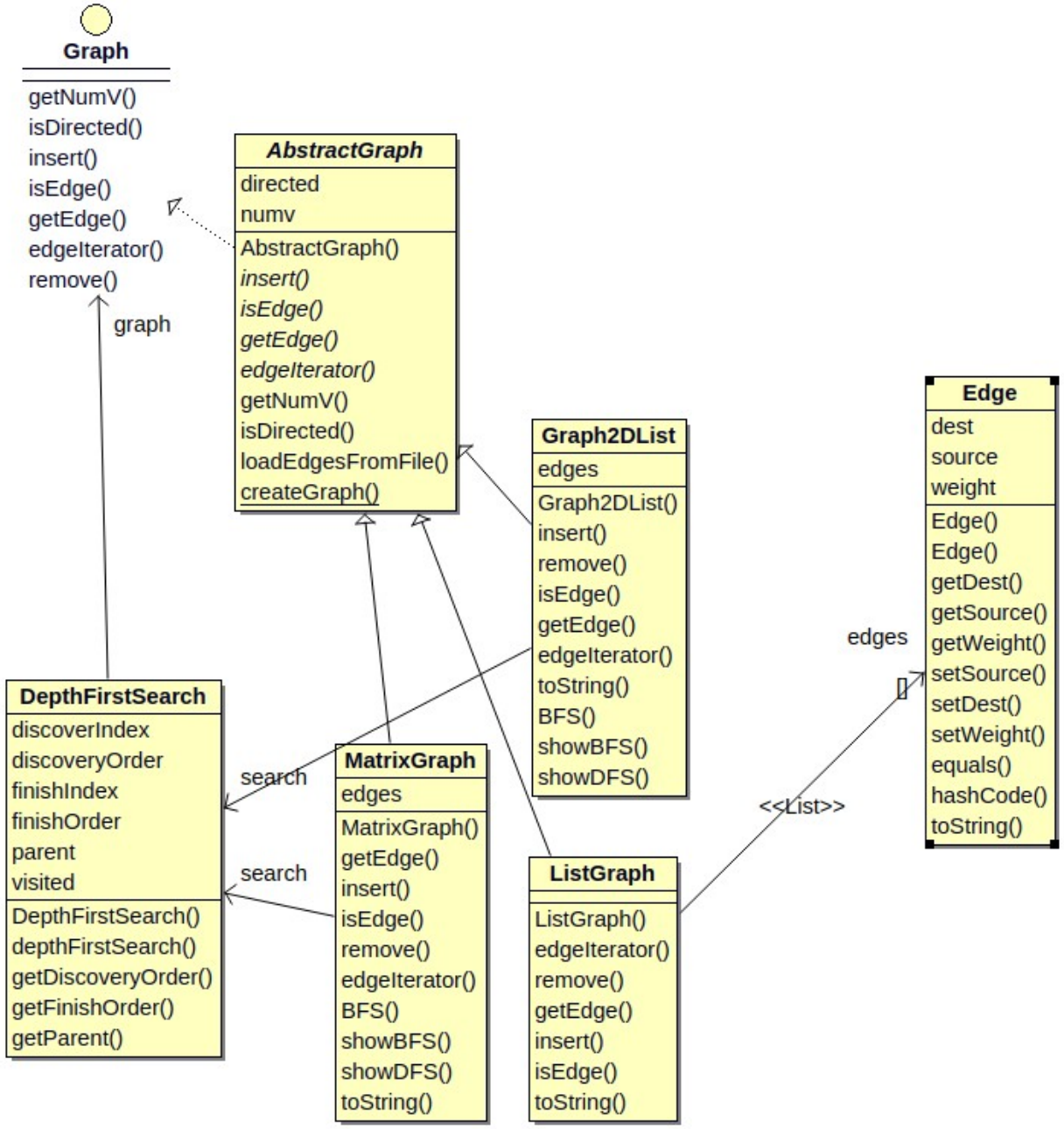
GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 4 Report

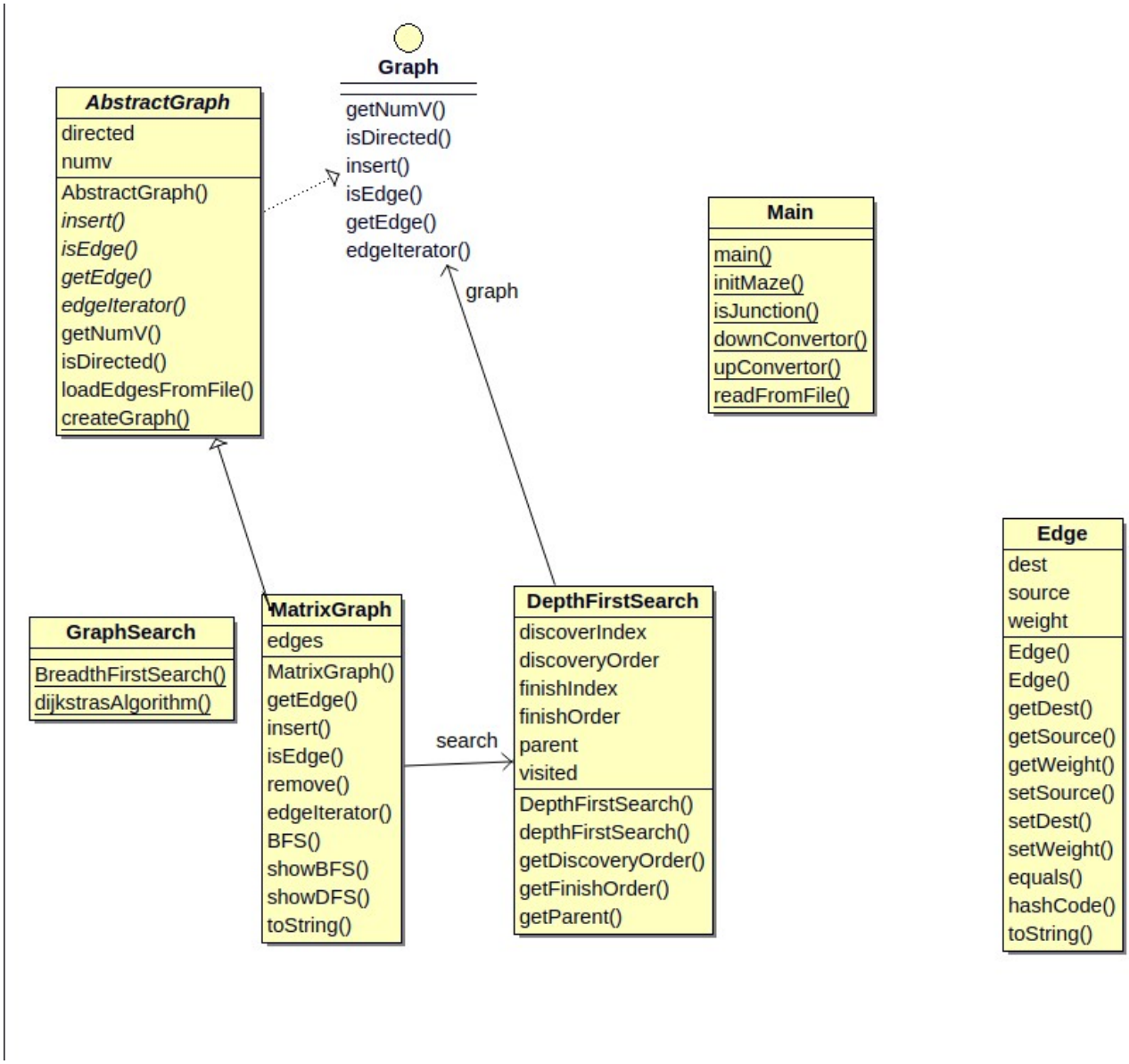
Muharrem Ozan Yeşiller
171044033

1) Class Diagrams

Part 2



Part 3


Graph
getNumV()
isDirected()
insert()
isEdge()
getEdge()
edgeIterator()
AbstractGraph
directed
numv
AbstractGraph()
insert()
isEdge()
getEdge()
edgeIterator()
getNumV()
isDirected()
loadEdgesFromFile()
createGraph()
graph
Main
main()
initMaze()
isJunction()
downConvertor()
upConvertor()
readFromFile()
Edge
dest
source
weight
Edge()
Edge()
getDest()
getSource()
getWeight()
setSource()
setDest()
setWeight()
equals()
hashCode()
toString()
GraphSearch
BreadthFirstSearch()
dijkstrasAlgorithm()
MatrixGraph
edges
MatrixGraph()
getEdge()
insert()
isEdge()
remove()
edgeIterator()
BFS()
showBFS()
showDFS()
toString()
search
DepthFirstSearch
discoverIndex
discoveryOrder
finishIndex
finishOrder
parent
visited
DepthFirstSearch()
depthFirstSearch()
getDiscoveryOrder()
getFinishOrder()
getParent()

Problem Solution Approach

Part 2

First of all, I had to translate the array list to show it with a two-dimensional list. I used the get method of the list for each operation that I accessed with the array with the array. Thus, it became a dynamic graph structure. Since the graf structure is concerned with what it shows, not how it is shown, this structure is not contrary to the graph theory. The class I have also implemented can make depth first and breadth first search calls. In addition to the implementations in the book, there is a deletion process. I kept what I added in the tests in an arraylist and successfully completed my tests by deleting as many vertex elements from that array list.

Part 3

I initialized the data from a file given in this section. We can make this structure weighted with shallow priority search. And when we use this structure, edges are created to reach the most optimized solution. Another detail in the initiliaze section is that if a structure like "java Main maze.txt" is used while keeping the file in the source folder and starting it, the program will run without errors.

Test Case

Part2

Insert undirected graph → Pass
Insert directed graph → Pass
Delete from undirected graph → Pass
Delete from directed graph → Pass
BFS for directed graph → Pass
BFS for undirected graph → Pass
DFS for directed graph → Pass
DFS for undirected graph → Pass
Converting Matrix graph to 2dlistgraph → Pass
(Directed and un directed)

Part 3

Shortest Path from file → Pass

Running commands and Result

```
Instance of Matrix Graph Directed:
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.0 1.0 0.0 0.0 1.0 0.0 1.0 0.0 0.0 0.0
0.0 0.0 1.0 1.0 0.0 0.0 1.0 0.0 1.0 0.0
0.0 0.0 0.0 0.0 0.0 1.0 0.0 1.0 0.0 0.0
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0
1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
0.0 1.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0


Instance of Matrix Graph UnDirected:
0.0 1.0 1.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0
1.0 1.0 0.0 0.0 0.0 1.0 0.0 1.0 0.0 0.0
1.0 0.0 0.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0
0.0 0.0 1.0 1.0 0.0 0.0 1.0 1.0 0.0 0.0
1.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0
0.0 1.0 1.0 0.0 1.0 0.0 0.0 1.0 1.0 1.0
0.0 0.0 1.0 1.0 0.0 0.0 0.0 1.0 1.0 1.0
0.0 1.0 1.0 1.0 0.0 1.0 1.0 0.0 0.0 0.0
1.0 0.0 1.0 0.0 0.0 1.0 1.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 1.0 1.0 1.0 0.0 0.0 0.0

Insert...
Graph2DList directed result:
Source: 0, Dest: 6, W: 1.0
Source: 2, Dest: 1, W: 1.0
Source: 2, Dest: 4, W: 1.0
Source: 2, Dest: 6, W: 1.0
Source: 3, Dest: 2, W: 1.0
Source: 3, Dest: 3, W: 1.0
Source: 3, Dest: 6, W: 1.0
Source: 3, Dest: 8, W: 1.0
Source: 4, Dest: 5, W: 1.0
Source: 4, Dest: 7, W: 1.0
Source: 5, Dest: 4, W: 1.0
Source: 6, Dest: 3, W: 1.0
Source: 6, Dest: 4, W: 1.0
Source: 6, Dest: 5, W: 1.0
Source: 7, Dest: 1, W: 1.0
Source: 7, Dest: 2, W: 1.0
Source: 8, Dest: 6, W: 1.0
Source: 9, Dest: 1, W: 1.0
Source: 9, Dest: 3, W: 1.0
Source: 9, Dest: 6, W: 1.0
```

```
Insert...
Graph2DList undirected result:
Source: 0, Dest: 1, W: 1.0
Source: 0, Dest: 2, W: 1.0
Source: 0, Dest: 4, W: 1.0
Source: 0, Dest: 8, W: 1.0
Source: 1, Dest: 1, W: 1.0
Source: 1, Dest: 5, W: 1.0
Source: 1, Dest: 7, W: 1.0
Source: 2, Dest: 3, W: 1.0
Source: 2, Dest: 4, W: 1.0
Source: 2, Dest: 5, W: 1.0
Source: 2, Dest: 6, W: 1.0
Source: 2, Dest: 7, W: 1.0
Source: 2, Dest: 8, W: 1.0
Source: 3, Dest: 2, W: 1.0
Source: 3, Dest: 3, W: 1.0
Source: 3, Dest: 6, W: 1.0
Source: 3, Dest: 7, W: 1.0
Source: 4, Dest: 2, W: 1.0
Source: 4, Dest: 5, W: 1.0
Source: 4, Dest: 9, W: 1.0
Source: 5, Dest: 1, W: 1.0
Source: 5, Dest: 2, W: 1.0
Source: 5, Dest: 4, W: 1.0
Source: 5, Dest: 7, W: 1.0
Source: 5, Dest: 8, W: 1.0
Source: 5, Dest: 9, W: 1.0
Source: 6, Dest: 2, W: 1.0
Source: 6, Dest: 3, W: 1.0
Source: 6, Dest: 7, W: 1.0
Source: 6, Dest: 8, W: 1.0
Source: 6, Dest: 9, W: 1.0
Source: 7, Dest: 1, W: 1.0
Source: 7, Dest: 2, W: 1.0
Source: 7, Dest: 3, W: 1.0
Source: 7, Dest: 5, W: 1.0
Source: 7, Dest: 6, W: 1.0
Source: 8, Dest: 2, W: 1.0
Source: 8, Dest: 5, W: 1.0
Source: 8, Dest: 6, W: 1.0
Source: 9, Dest: 4, W: 1.0
Source: 9, Dest: 5, W: 1.0
Source: 9, Dest: 6, W: 1.0
```

```
DFS result for directed graph2dlist:
Discovery and finish order
0 1
6 5
3 7
2 4
1 2
4 8
5 3
7 6
8 0
Parent:
[0]: -1
[1]: 2
[2]: 3
[3]: 6
[4]: 2
[5]: 4
[6]: 0
[7]: 4
[8]: 3
[9]: -1

DFS result for undirected graph2dlist:
Discovery and finish order
0 7
1 8
5 4
2 9
3 6
6 3
7 2
8 5
9 1
4 0
Parent:
[0]: -1
[1]: 0
[2]: 5
[3]: 2
[4]: 9
[5]: 1
[6]: 3
[7]: 6
[8]: 6
[9]: 6
```

```
BFS result for undirected graph2dlist:
[0]: -1
[1]: 0
[2]: 0
[3]: 2
[4]: 0
[5]: 1
[6]: 2
[7]: 1
[8]: 0
[9]: 4

Removed piece of vertex/2 from graphlist2d directed
Source: 0, Dest: 6, W: 1.0
Source: 2, Dest: 1, W: 1.0
Source: 2, Dest: 4, W: 1.0
Source: 2, Dest: 6, W: 1.0
Source: 3, Dest: 2, W: 1.0
Source: 3, Dest: 3, W: 1.0
Source: 3, Dest: 6, W: 1.0
Source: 3, Dest: 8, W: 1.0
Source: 4, Dest: 5, W: 1.0
Source: 4, Dest: 7, W: 1.0
Source: 5, Dest: 4, W: 1.0
Source: 6, Dest: 3, W: 1.0
Source: 6, Dest: 4, W: 1.0
Source: 6, Dest: 5, W: 1.0
Source: 7, Dest: 2, W: 1.0
Source: 8, Dest: 6, W: 1.0
Source: 9, Dest: 3, W: 1.0
Source: 9, Dest: 6, W: 1.0
```

```
Removed piece of vertex/2 from graphlist2d undirected
Source: 0, Dest: 1, W: 1.0
Source: 0, Dest: 2, W: 1.0
Source: 0, Dest: 4, W: 1.0
Source: 0, Dest: 8, W: 1.0
Source: 1, Dest: 1, W: 1.0
Source: 1, Dest: 5, W: 1.0
Source: 1, Dest: 7, W: 1.0
Source: 2, Dest: 3, W: 1.0
Source: 2, Dest: 4, W: 1.0
Source: 2, Dest: 6, W: 1.0
Source: 2, Dest: 7, W: 1.0
Source: 2, Dest: 8, W: 1.0
Source: 3, Dest: 3, W: 1.0
Source: 3, Dest: 6, W: 1.0
Source: 3, Dest: 7, W: 1.0
Source: 4, Dest: 5, W: 1.0
Source: 4, Dest: 9, W: 1.0
Source: 5, Dest: 1, W: 1.0
Source: 5, Dest: 2, W: 1.0
Source: 5, Dest: 4, W: 1.0
Source: 5, Dest: 7, W: 1.0
Source: 5, Dest: 8, W: 1.0
Source: 5, Dest: 9, W: 1.0
Source: 6, Dest: 2, W: 1.0
Source: 6, Dest: 3, W: 1.0
Source: 6, Dest: 7, W: 1.0
Source: 6, Dest: 8, W: 1.0
Source: 6, Dest: 9, W: 1.0
Source: 7, Dest: 1, W: 1.0
Source: 7, Dest: 2, W: 1.0
Source: 7, Dest: 3, W: 1.0
Source: 7, Dest: 5, W: 1.0
Source: 7, Dest: 6, W: 1.0
Source: 8, Dest: 2, W: 1.0
Source: 8, Dest: 5, W: 1.0
Source: 8, Dest: 6, W: 1.0
Source: 9, Dest: 4, W: 1.0
Source: 9, Dest: 5, W: 1.0
Source: 9, Dest: 6, W: 1.0
```

Part 3

```
rhasepy@MRX:~/Desktop/MazeSolver/src$ java Main maze.txt
[0]: -1
[1]: 0
[2]: 0
[3]: 0
[4]: 0
[5]: 0
[6]: 1
[7]: 1
[8]: 1
[9]: 1
[10]: 1
[11]: 1
[12]: 1
[13]: 1
[14]: 1
[15]: 1
[16]: 1
[17]: 1
[18]: 1
[19]: 1
[20]: 1
[21]: 1
[22]: 1
[23]: 13
The shortest path is:
1
13
23
```