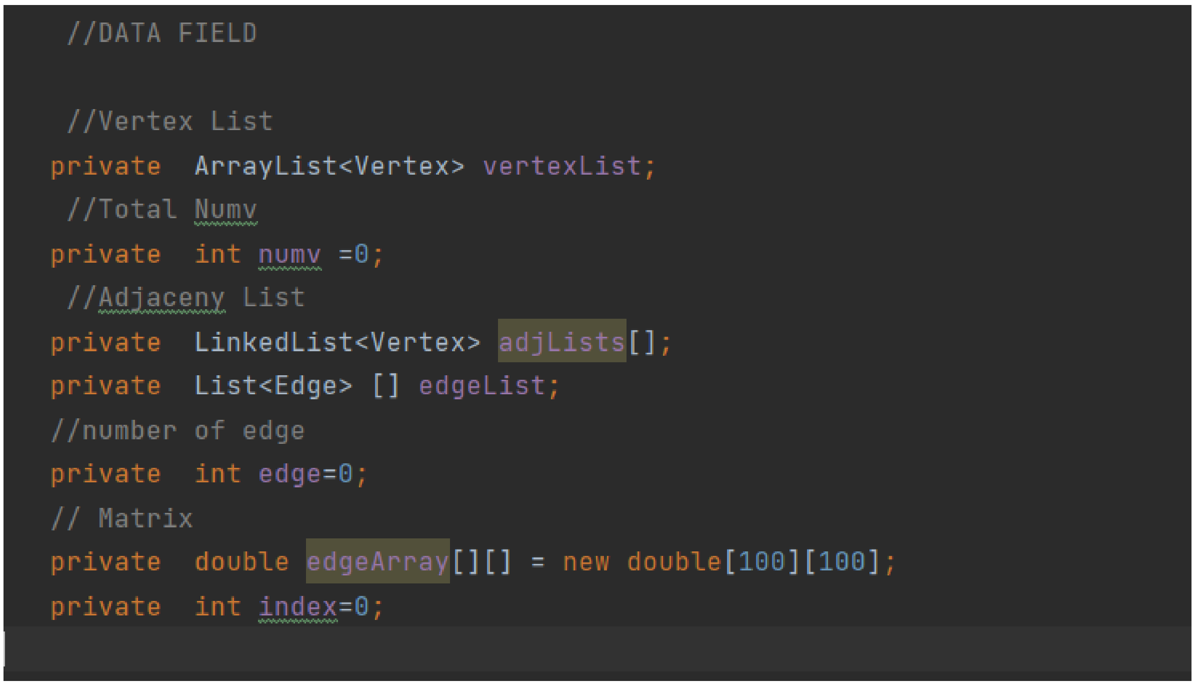
GTU Department of Computer Engineering

CSE 222/505 - SPRING 2022 HOMEWORK 8 REPORT

MUHAMMET ÇAĞRI YILMAZ
1901042694

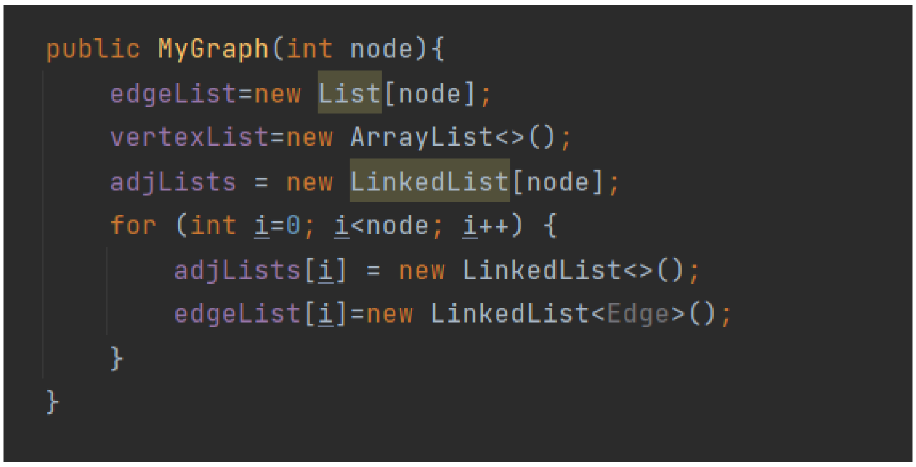
System Requirements

In MyGraph Class

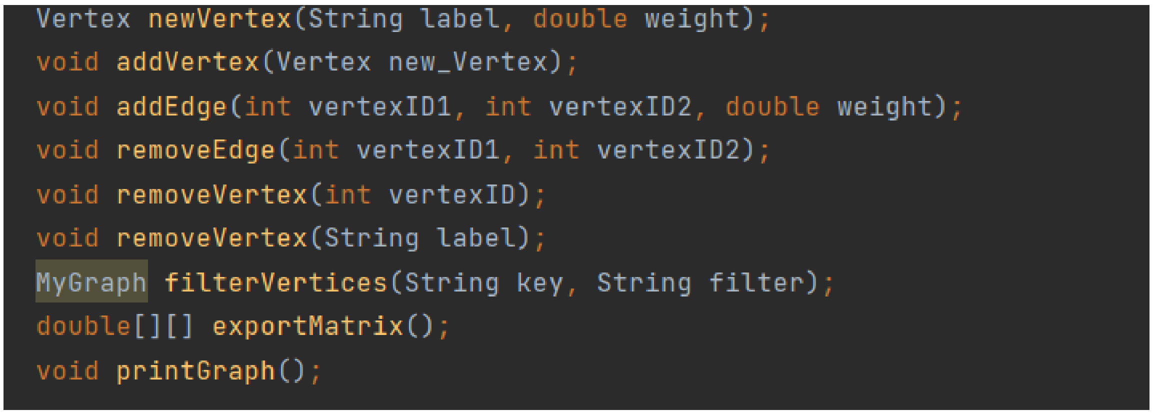


```
  //DATA FIELD

  //Vertex List
 private  ArrayList<Vertex> vertexList;
  //Total Numv
 private  int numv =0;
  //Adjaceny List
 private  LinkedList<Vertex> adjLists[];
 private  List<Edge> [] edgeList;
 //number of edge
 private  int edge=0;
 // Matrix
 private  double edgeArray[][] = new double[100][100];
 private  int index=0;
```

Constructor



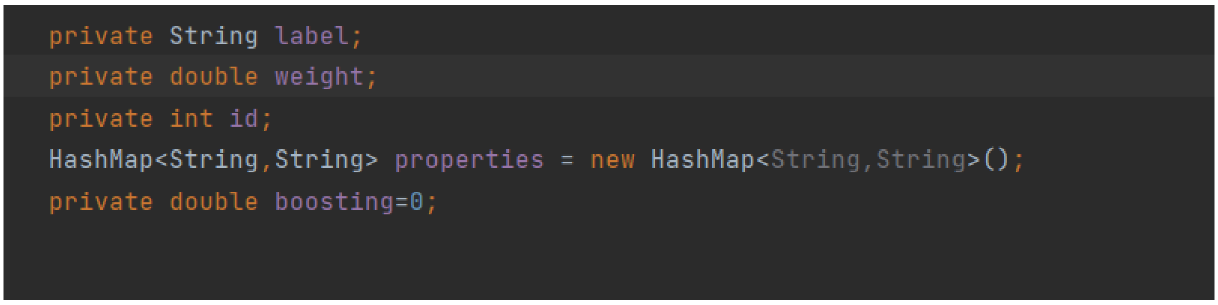
```
public MyGraph(int node){
    edgeList=new List[node];
    vertexList=new ArrayList<>();
    adjLists = new LinkedList[node];
    for (int i=0; i<node; i++) {
        adjLists[i] = new LinkedList<>();
        edgeList[i]=new LinkedList<Edge>();
    }
}
```



```
Vertex newVertex(String label, double weight);
void addVertex(Vertex new_Vertex);
void addEdge(int vertexID1, int vertexID2, double weight);
void removeEdge(int vertexID1, int vertexID2);
void removeVertex(int vertexID);
void removeVertex(String label);
MyGraph filterVertices(String key, String filter);
double[][] exportMatrix();
void printGraph();
```

All methods that we need to implements are in an Interface.

Vertex Class

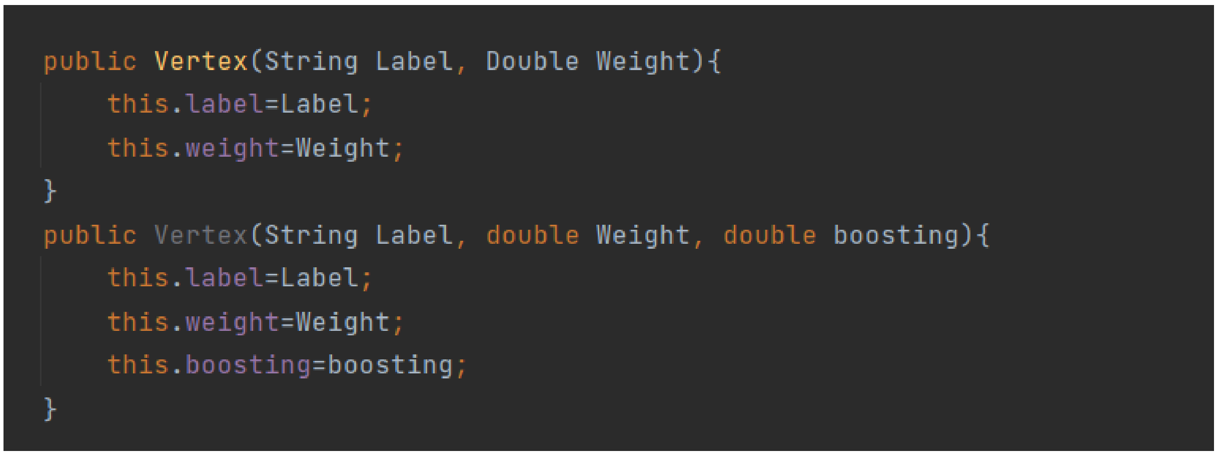


```
private String label;
private double weight;
private int id;
HashMap<String,String> properties = new HashMap<String,String>();
private double boosting=0;
```

Id variable holds the both id and index.

Properties holds the properties of vertex.

There is a variable that’s called boosting. This variable is for 3rd question



```
public Vertex(String Label, Double Weight){
    this.label=Label;
    this.weight=Weight;
}
public Vertex(String Label, double Weight, double boosting){
    this.label=Label;
    this.weight=Weight;
    this.boosting=boosting;
}
```

There are 2 constructors. We need to second one to put vertex class for 3rd question.

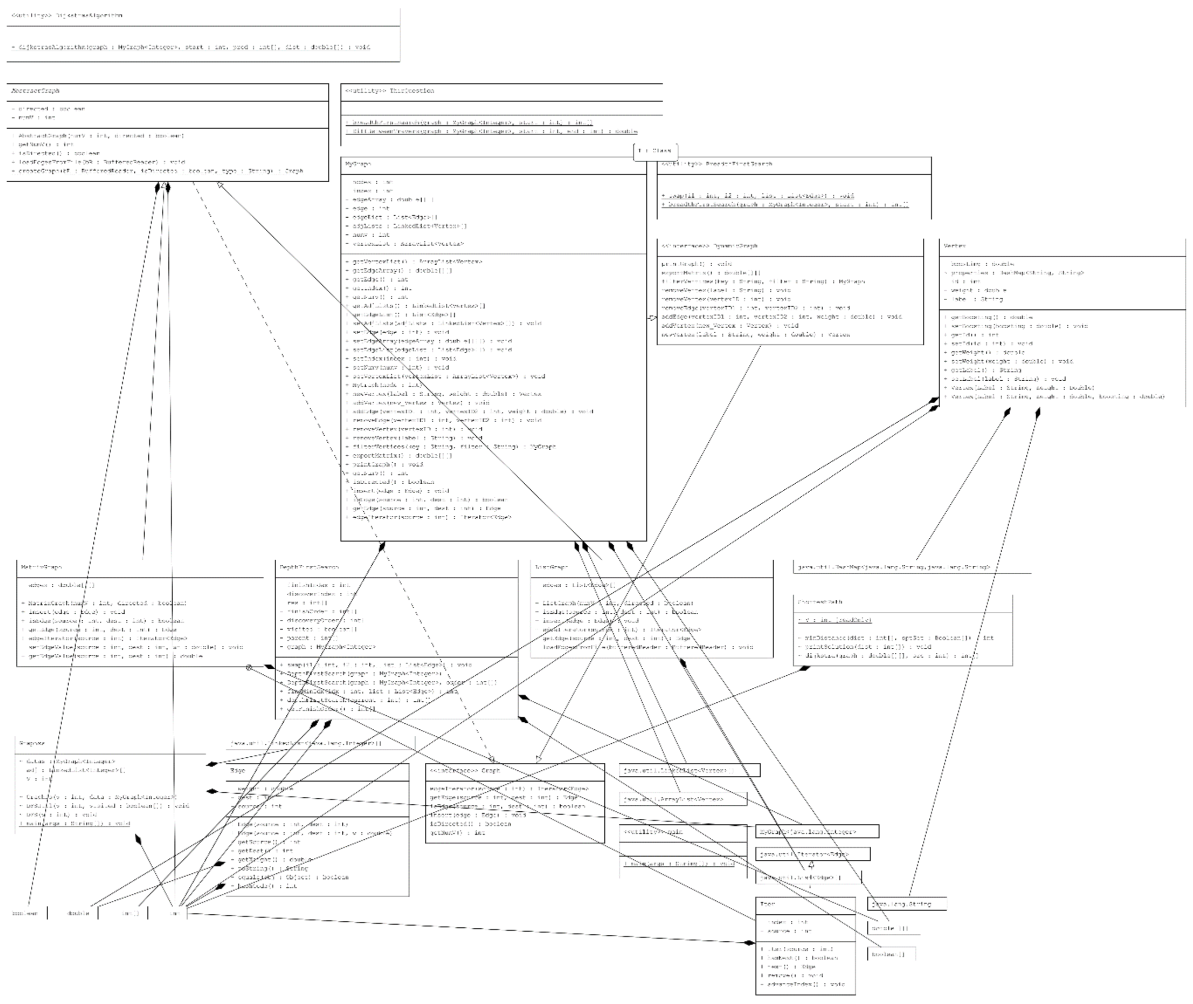
Problem Solutions Approach

For 1st question, I read Graph section our lecture book. After that I implemented all things that we require from the book. I slog on during adjency list. Even our lecturer said that you don’t need to use Edge class, I used it. Thanks to that, I did my assignment easier.

For 2nd question, this is the hardest part for me. Because in assignment, lecturer wants that We find the shortest part. I examined each edge and vertex. BFS we finished each level and that’s why this part is hard. DFS part is easier than BFS.

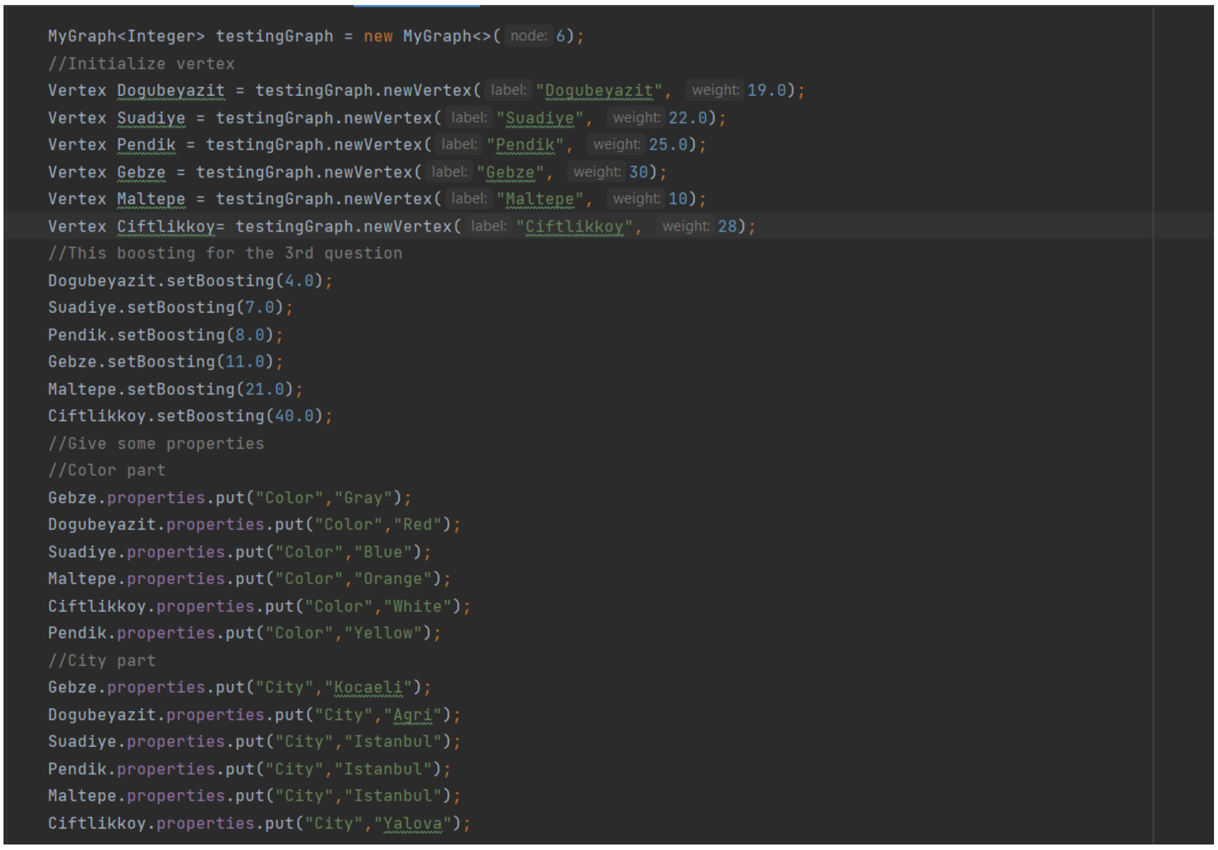
For 3rd question, I watched Dijkstra’s Algorithm on YouTube and take code from lecture book. However, I need to modify somethings. Boosting value is important and I compare the boosting value for each vertex.

CLASS DIAGRAM

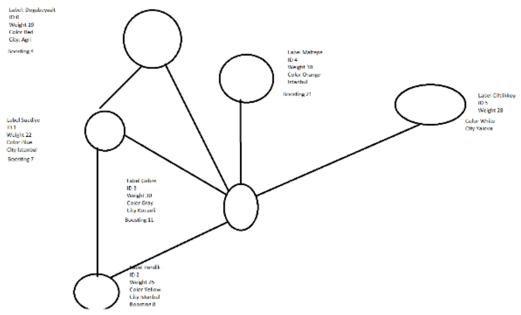


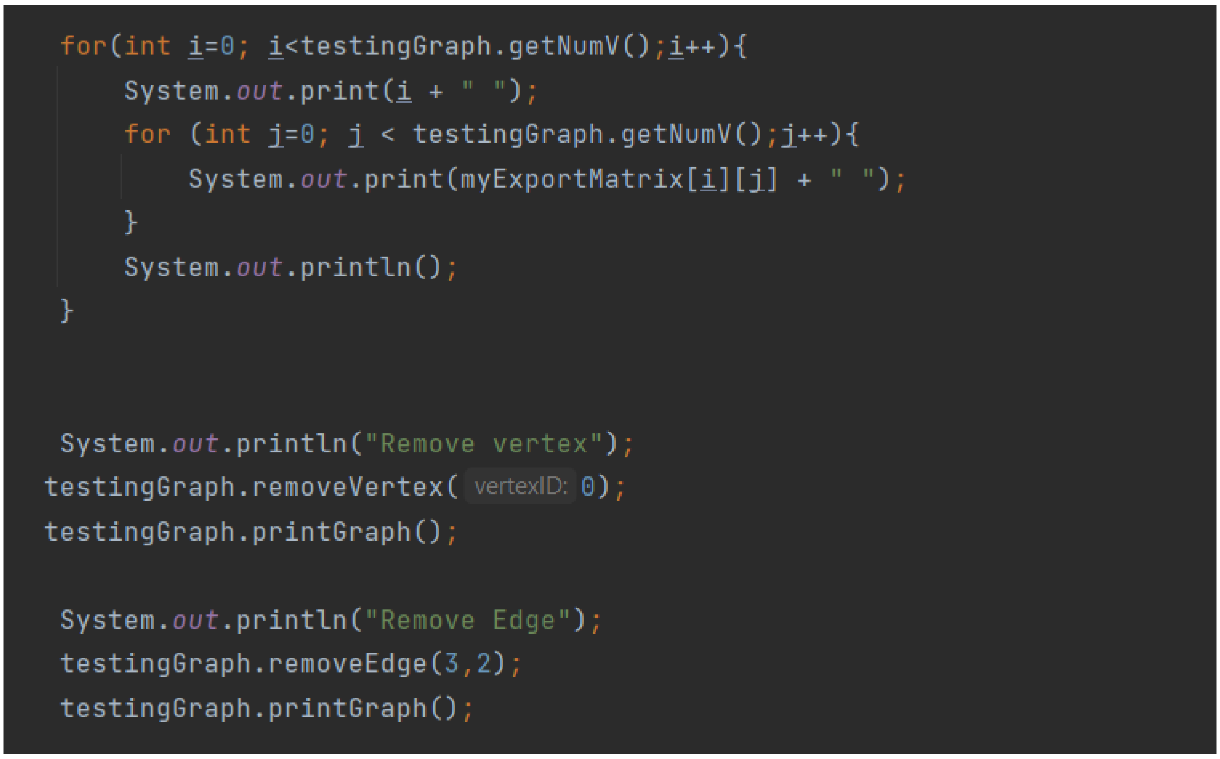
Test Cases

For 1st question



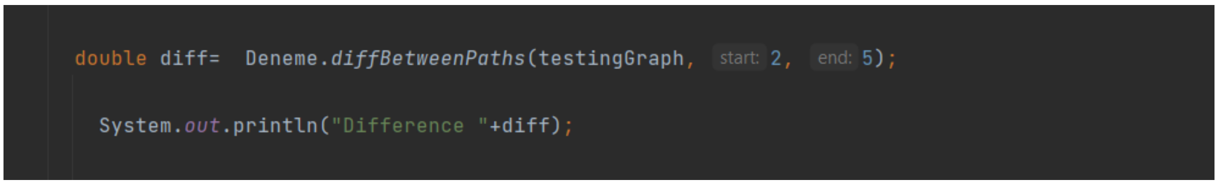
```
MyGraph<Integer> testingGraph = new MyGraph<>( node: 6);
//Initialize vertex
Vertex Dogubeyazit = testingGraph.newVertex( label: "Dogubeyazit",  weight: 19.0);
Vertex Suadiye = testingGraph.newVertex( label: "Suadiye",  weight: 22.0);
Vertex Pendik = testingGraph.newVertex( label: "Pendik",  weight: 25.0);
Vertex Gebze = testingGraph.newVertex( label: "Gebze",  weight: 30);
Vertex Maltepe = testingGraph.newVertex( label: "Maltepe",  weight: 10);
Vertex Ciftlikkoy= testingGraph.newVertex( label: "Ciftlikkoy",  weight: 28);
//This boosting for the 3rd question
Dogubeyazit.setBoosting(4.0);
Suadiye.setBoosting(7.0);
Pendik.setBoosting(8.0);
Gebze.setBoosting(11.0);
Maltepe.setBoosting(21.0);
Ciftlikkoy.setBoosting(40.0);
//Give some properties
//Color part
Gebze.properties.put("Color","Gray");
Dogubeyazit.properties.put("Color","Red");
Suadiye.properties.put("Color","Blue");
Maltepe.properties.put("Color","Orange");
Ciftlikkoy.properties.put("Color","White");
Pendik.properties.put("Color","Yellow");
//City part
Gebze.properties.put("City","Kocaeli");
Dogubeyazit.properties.put("City","Agri");
Suadiye.properties.put("City","Istanbul");
Pendik.properties.put("City","Istanbul");
Maltepe.properties.put("City","Istanbul");
Ciftlikkoy.properties.put("City","Yalova");
```





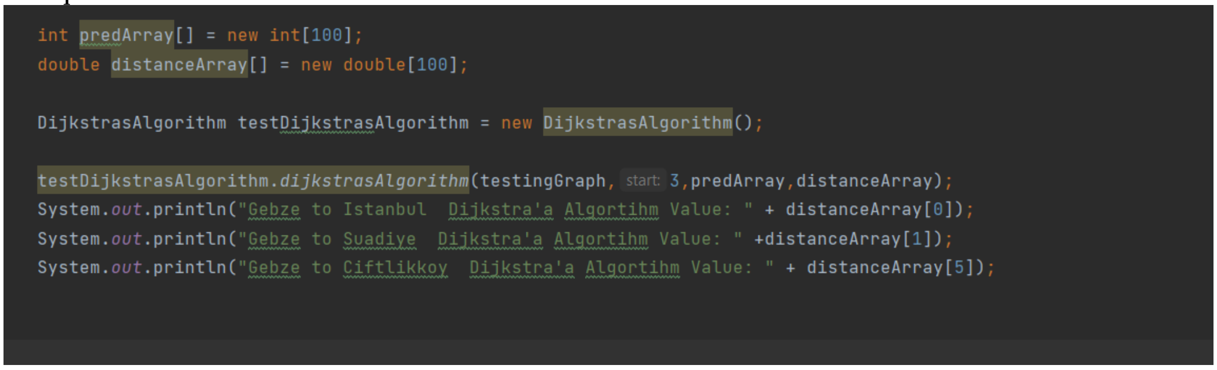
```
for(int i=0; i<testingGraph.getNumV();i++){
    System.out.print(i + " ");
    for (int j=0; j < testingGraph.getNumV();j++){
        System.out.print(myExportMatrix[i][j] + " ");
    }
    System.out.println();
}


System.out.println("Remove vertex");
testingGraph.removeVertex( vertexID: 0);
testingGraph.printGraph();

System.out.println("Remove Edge");
testingGraph.removeEdge(3,2);
testingGraph.printGraph();
```

2nd Question



```
double diff=  Deneme.diffBetweenPaths(testingGraph, start: 2, end: 5);

  System.out.println("Difference "+diff);
```

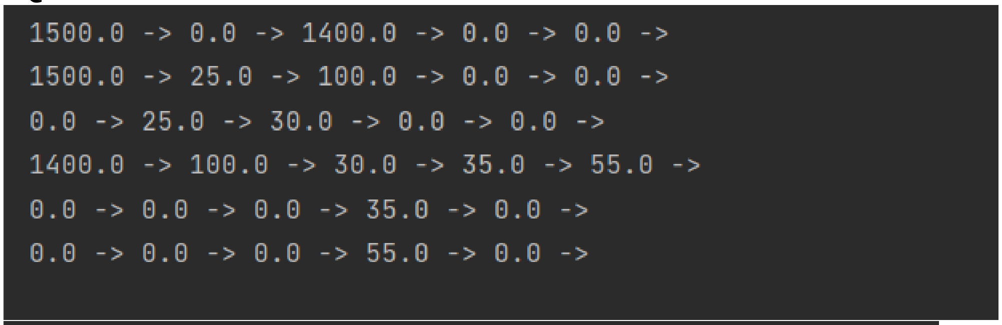
3rd question



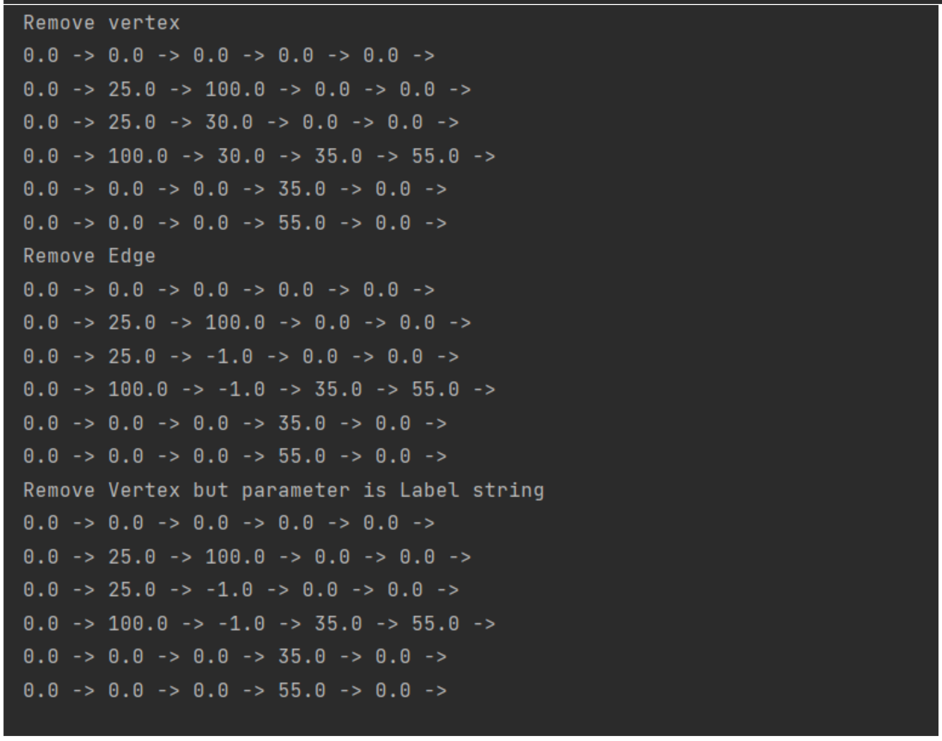
```
int predArray[] = new int[100];
double distanceArray[] = new double[100];

DijkstrasAlgorithm testDijkstrasAlgorithm = new DijkstrasAlgorithm();

testDijkstrasAlgorithm.dijkstrasAlgorithm(testingGraph, start: 3,predArray,distanceArray);
System.out.println("Gebze to Istanbul  Dijkstra'a Algortihm Value: " + distanceArray[0]);
System.out.println("Gebze to Suadiye  Dijkstra'a Algortihm Value: " +distanceArray[1]);
System.out.println("Gebze to Ciftlikkoy  Dijkstra'a Algortihm Value: " + distanceArray[5]);
```

Test Results

Question 1

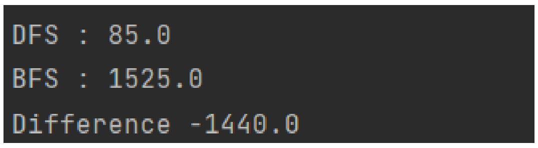


```
1500.0 -> 0.0 -> 1400.0 -> 0.0 -> 0.0 ->
1500.0 -> 25.0 -> 100.0 -> 0.0 -> 0.0 ->
0.0 -> 25.0 -> 30.0 -> 0.0 -> 0.0 ->
1400.0 -> 100.0 -> 30.0 -> 35.0 -> 55.0 ->
0.0 -> 0.0 -> 0.0 -> 35.0 -> 0.0 ->
0.0 -> 0.0 -> 0.0 -> 55.0 -> 0.0 ->
```



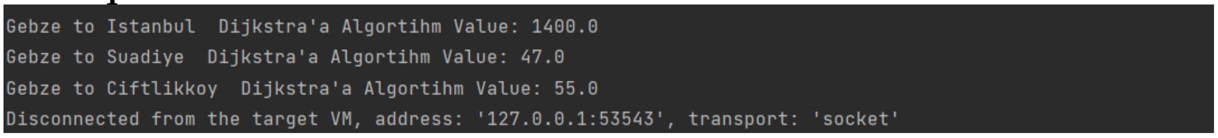
```
Remove vertex
0.0 -> 0.0 -> 0.0 -> 0.0 -> 0.0 ->
0.0 -> 25.0 -> 100.0 -> 0.0 -> 0.0 ->
0.0 -> 25.0 -> 30.0 -> 0.0 -> 0.0 ->
0.0 -> 100.0 -> 30.0 -> 35.0 -> 55.0 ->
0.0 -> 0.0 -> 0.0 -> 35.0 -> 0.0 ->
0.0 -> 0.0 -> 0.0 -> 55.0 -> 0.0 ->
Remove Edge
0.0 -> 0.0 -> 0.0 -> 0.0 -> 0.0 ->
0.0 -> 25.0 -> 100.0 -> 0.0 -> 0.0 ->
0.0 -> 25.0 -> -1.0 -> 0.0 -> 0.0 ->
0.0 -> 100.0 -> -1.0 -> 35.0 -> 55.0 ->
0.0 -> 0.0 -> 0.0 -> 35.0 -> 0.0 ->
0.0 -> 0.0 -> 0.0 -> 55.0 -> 0.0 ->
Remove Vertex but parameter is Label string
0.0 -> 0.0 -> 0.0 -> 0.0 -> 0.0 ->
0.0 -> 25.0 -> 100.0 -> 0.0 -> 0.0 ->
0.0 -> 25.0 -> -1.0 -> 0.0 -> 0.0 ->
0.0 -> 100.0 -> -1.0 -> 35.0 -> 55.0 ->
0.0 -> 0.0 -> 0.0 -> 35.0 -> 0.0 ->
0.0 -> 0.0 -> 0.0 -> 55.0 -> 0.0 ->
```

Question 2



```
DFS : 85.0
BFS : 1525.0
Difference -1440.0
```

For question 3



```
Gebze to Istanbul  Dijkstra'a Algortihm Value: 1400.0
Gebze to Suadiye  Dijkstra'a Algortihm Value: 47.0
Gebze to Ciftlikkoy  Dijkstra'a Algortihm Value: 55.0
Disconnected from the target VM, address: '127.0.0.1:53543', transport: 'socket'
```

Thanks for everything this semester.