GIT Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework #8 Report

171044073
Berkan AKIN

1. Problem Definetion

Question 1)
A program should be written on java that will run on all computers. A function will be written for the program. 8 specified functions should be implemented. Graph interface should be written. However, the Mygraph class must be written. A vertex must have an index (ID), a label, and a weight. The vertices may have user-defined additional properties (Vertex class should be generic)

Question 2)
Write a method that takes a MyGraph object as a parameter and performs BFS and DFS traversals. Method should calculate difference between 2 types of travel. If there are more than one alternative to access a vertex at a specific level during the BFS, the shortest alternative should be considered

Question 3)
Write a method that takes a MyGraph object and a vertex as a parameter to perform a modified version of Dijkstra's Algorithm for calculating the shortest paths from the
given vertex to all other vertices in the graph. In this modified version, the algorithm considers boosting value of the vertices in addition to the edge weights.

2. System Requirements

Question 1)
The system should have libraries and java virtual machine to run the necessary functions. the desired program should be able to run without errors

3) Problem Solution Approach
Regarding my system's requirements and problems, i created a container class
to keep and modift the data easily. Then I was able to set up a hierarchy and find
a solution, by correctly determining the class relationships and the ease provided
by my container class.
PROBLEM SOLUTION APPROACH My Problem solution steps are;
1. – Specify the problem requirements
2. – Analyze the problem
3. – Design an algorithm and Program
4. – Implement the algorithm
5. – Test and verify the program
6. – Maintain and update the program
I create and test each class separately. Then I combine these classes
according to the problem and test them.
Question 1)
First of all, we must understand what the problem is. After defining the problem, we must implement the necessary functions. First, the necessary interfaces must be implemented. Classes must be written and necessary functions must be written. After the functions are written, the test phase should be passed and the functions should be tested.
Question 2)

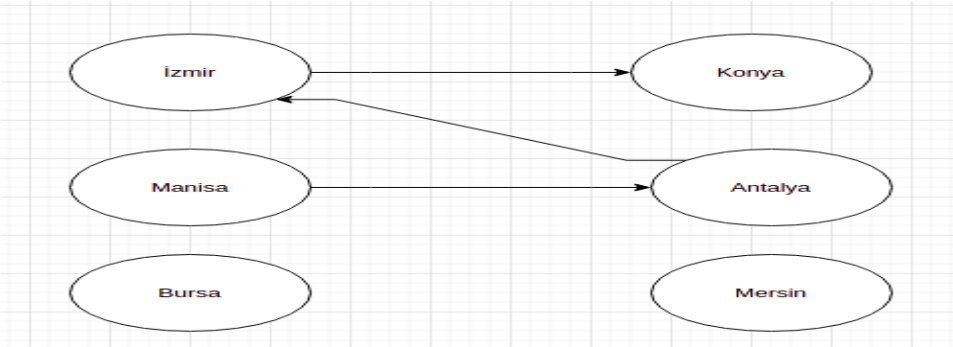
Question 3)

4) Test cases

Question 1)

Creating Graph class, creating vertex, adding vertex to mxgraph class and creating edge operations are done.

```
public static void main(String[] args) {
    MyGraph mg = new MyGraph(10);
    Vertex vt = mg.newVertex("manisa   ", 30);  // id 0
    mg.addVertex(vt);
    vt = mg.newVertex("izmir    ", 50);         // id 1
    mg.addVertex(vt);

    vt = mg.newVertex("konya    ", 70);         // id 2
    mg.addVertex(vt);

    vt = mg.newVertex("antalya  ", 40);     // id 3
    mg.addVertex(vt);

    vt = mg.newVertex("mersin   ", 100);        // id 4
    mg.addVertex(vt);

    vt = mg.newVertex("Bursa    ", 50);         // id 5
    mg.addVertex(vt);

    mg.addEdge(1, 2, 30);
    mg.addEdge(0,3,40);
    mg.addEdge(3, 1, 40);
```



Remove Edge Remove Vertex test

```
mg.removeEdge(1, 3);
mg.removeVertex(0);
mg.removeVertex("antalya");
```

4) Test and Results

Question 1)

Test	Expected	Output	Results
			Pass
			Pass

Test	Expected	Output	Results
			Pass
			Pass
			Pass
			Pass
			Pass
			Pass
			

5)Run Time Complexity Analysis

Question 1)

Function	Best Case	Worst Case	Average Case
	θ(1)	O(1)	O(1)
	θ(1)	O(n)	θ(n)

Function	Best Case	Worst Case	Average Case
	θ(1)	θ(n)	θ(n)
	θ(1)	θ(n)	θ(n)
	θ(1)	θ(n)	θ(n)
	θ(1)	θ(n)	θ(logn)

Function	Best Case	Worst Case	Average Case
	θ(1)	θ(n)	θ(logn)
	$\theta(n^2)$	$\theta(n^2)$	$\theta(n^2)$
	θ(1)	O(n)	O(n)