

Information:

Giray Berk Kuşhan Section 04

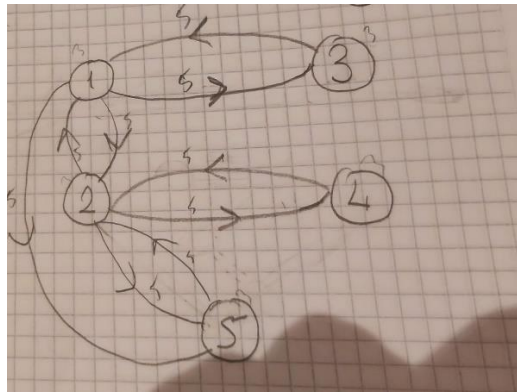
10889878942

Homework 1

Q1:

Problem Statement and Code Design:

In question 1 I must use graph implementations according to the airline company. In this question there are 5 different vertices and rotations from each other. So, I occurred Graph class and its methods in order to find undirected rotations by using Graph implementation. Then, created main class and read inputs and use objects which are in Graph class.



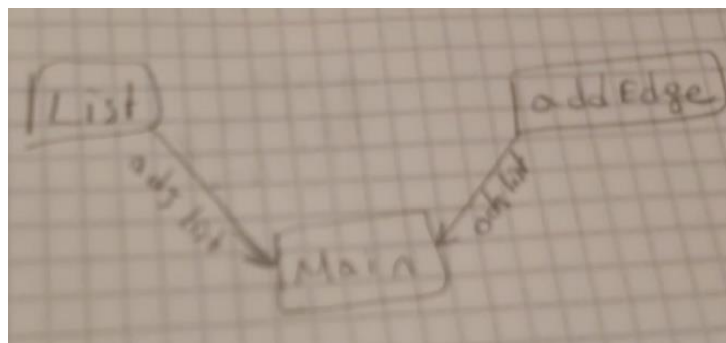
This is the chart of the question 1.

Implementation and Functionality:

`addEdge(int u, int v)`: This code defines method for undirected edge between vertices with an integer in graph.

`getNeighbors(int u)`: This code is used for returns a list of neighbours a vertex u in the graph class.

`Main()`: reads inputs by using scanner class and calculate the shortest path between two cities in a graph using Dijkstra's algorithm.



Testing:

In testing part. I created additional 2 tests. First Test Case controlled inputs as 0 positive and negative scenarios. Second Test Case controlled vertices and edges. As a result of the 2 test cases. Codes is run clearly but in negative test case program gives an error because inputs were not controlled for negativity.

Final Assessments:

Completing this assignment is a little bit hard for me because there is a little confusion while I was reading question because I could not understand clearly what question wanted to me and creating a chart before starting programming is little complicated.

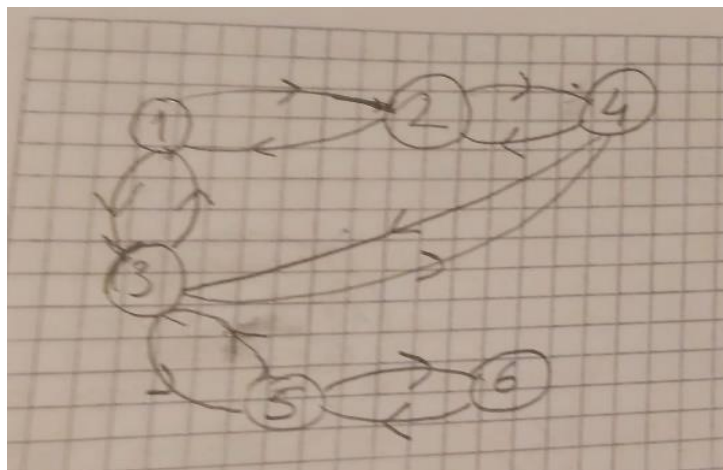
The hardest part is writing a code after creating graph chart.

I learned how to I use graph algorithm and how I create graph methods by using graph algorithms.

Q2:

Problem Statement and Code Design:

In question 2 I must use graph implementations according to organize a tour ship company and there are N Islands and M paths between these islands. Also, edges are undirected. So, I, write a java program which is useful for this assignment. I write Depth First Search algorithm in order to reach true outputs. Also I used Node class and HashMap in this part of the assignment.



This is the chart of the question 2.

Implementation and Functionality:

`addEdge()`: This code defines method for undirected edge between vertices with an integer in graph.

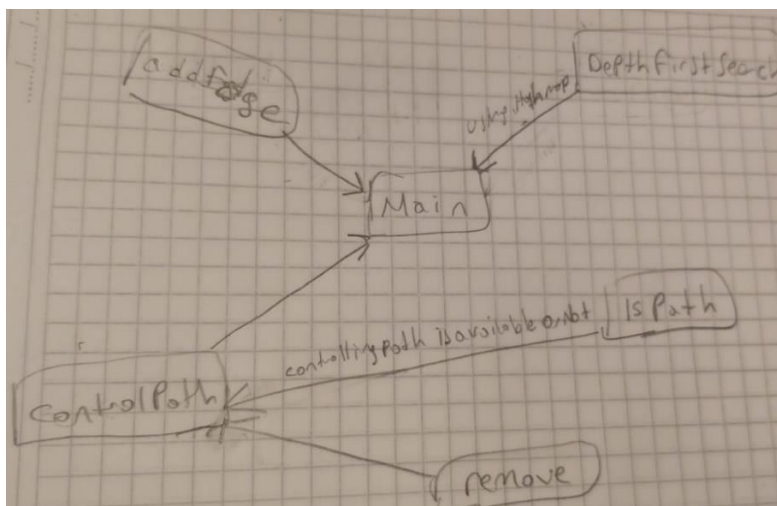
`Remove()`: Defines Iterator interface and controlled Exception as unsupported and `NoSuchElementException`.

`isPath(Graph graph, int index)`: Controlled vertex and adjacent in indexes whether marked or unmarked.

`ControlPath(Graph graph, int StartIndex, int FinalIndex)`: This method is finds all path by using recursion between start and end indexes. Also it collect them in hashMap.

`DepthFirstSearch(Graph G, int StartIndex)`: starts DFS object and sets the initial value for searching.

`Main()`: create graph object and performs DFS on the graph, and use print paths.



Testing:

In testing part, I created an additional test in order to examine zero, negative and positive values. So, there was a same error with question 1 because program does not control in case for zero value. Also, controlled for multiple paths for each path are valid. Therefore, I used different test for that. As a result, there is no checking for if there is a rotation the smallest path.

Final Assessments:

The question 2 is harder and longer than question 1 because there are many different algorithms are used in that part such as DFS, Hash etc. the hardest part is that using DFS algorithms to find true path. Also, I learned many algorithm features in this part of the assignment. The efficient part is that using DFS algorithm.