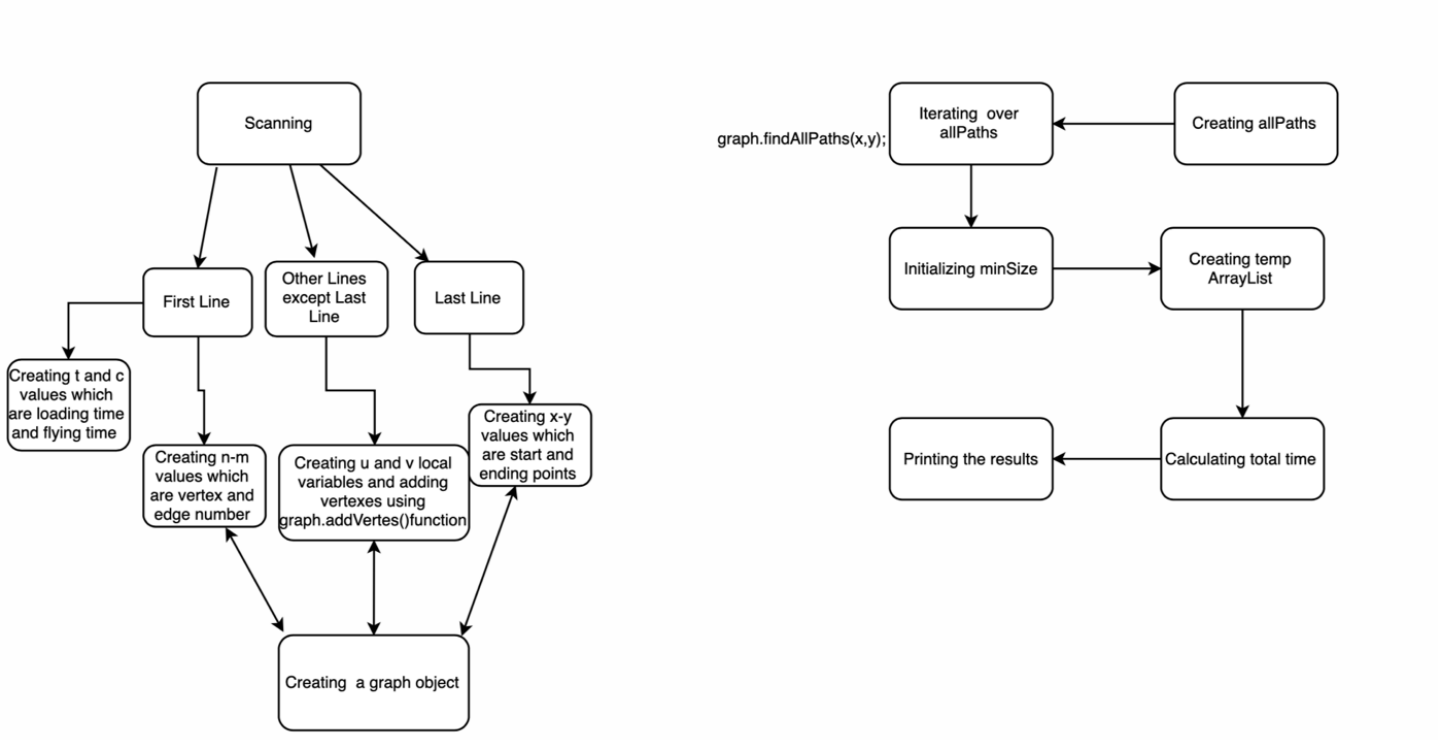
Sarp Arslan CMPE 224- SEC 03
11458145526 Assignment /HW-1
PA REPORT

Problem Statement and Code Design for Question 1

With this assignment, I created which finds the lexicographically smallest path which will take the minimum amount of time required to move from city X to city Y.



Implementation, Functionality

How the program works?

1- **Scanning** : The program scans all the input values with order. In the first input line, program initialize vertex numbers, edge numbers, loading time and flying time, starting and ending points.
2- **Creating a Graph Objcect :** The program creates a graph object, creates vertices, edges and start points for depth first search.
3- **Depth First Search :** Program is implementing a depth first search (dfs) algorithm to find all possible paths between a given start node and a destination node in a graph. The dfs algorithm is implemented recursively.
4- **Iteration :** Program iterates over allPaths and initializing minSize.

5- **Creating ArrayLists :** Program creates temp ArrayList
6- **Display** : The program prints the results after sorting the ArrayList.

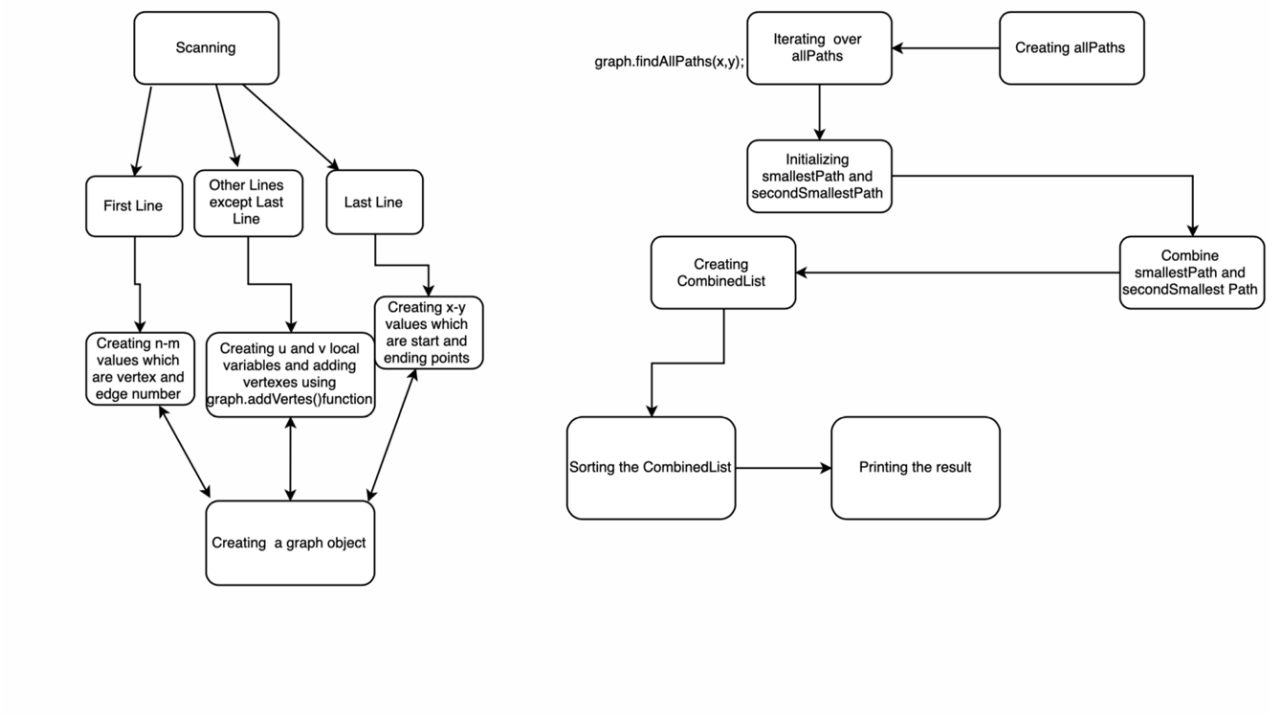
Testing

-There are 3 test cases, in the input, it gives the vertices size and prints until the last line. I printed graph's condition totalTime and counter information and leave it in the command line. The program works with different inputs which includes different edge and vertices sizes. In addition to that in all test cases program calculates the time without any mistake because the loading time is the same for all airports and it can caluclate accordingly to this information.

Problem Statement and Code Design for Question 2

With this assignment, I created a program that organizes a ship tour starts from island X and include island Y and return to island X from different path.

Implementation, Functionality



How the program works?

1-) **Scanning** : The program scanns all the input values with order. In the first input line, program initialize vertex numbers and edge numbers,starting and ending points.

2-) **Creating a Graph Object :** The program creates a graph object, creates vertices, edges and start points for depth first search.

3-) **Depth First Search :** Program is implementing a depth first search (dfs) algorithm to find all possible paths between a given start node and a destination node in a graph. The dfs algorithm is implemented recursively.

4-) **Iterating :** Program iterates over allPaths and initializing smallestPath and secondSmallestPath.

5-)**Combining two lists :** After that program combines smallestPath and secondSmallestPath and creates CombinedList.

6-)**Sorting and Printing the result :** The program sorts CombinedList and prints the result.

Testing

There are 3 test cases, while testing the code I printed the current graph situation in order to check whether graph is implementing correct or not. In addition to that I also checked the all posible paths. After that I also checked the desirable paths whose are the shortest pats.

FINAL ASSESSMENTS

With the help of this homework, I understand the working princible of inderected graphs, I learned add edges and since I learned how to connect vertices the information I learned in the classes is now more understandable. Because I learned how to implement graph in the code.