FAST NATIONAL UNIVERSITY

****

SUBJECT: Data Structures

Project(Airline Reservation System)

Group# A5

Section: A

Group Members:

Talha Mustafa(18i-0573)

Hanzala Ahmed Khan(18i-518)

Muhammad Asim(18i-0508)

**Algorithm Analysis used in Airline Reservation System**

This document discusses the data structure used in this project. We have tried to implement the best possible data structure to make sure we can reduce the time complexity as much as possible. We have discussed everything before implementing to make sure we don’t face any problem while implementing.

First, we have created a graph using Adjacency Matrix and a class Route Characteristics to store data based on our requirements. The main goal is to generate the Adjacency matrix so we can implement the algorithms in the simplest way.

**/x**



string date;

string fromTime;

string toTime;

int cost;

string airlineChoice;

string location;

List<RouteCharacteristics> destination;

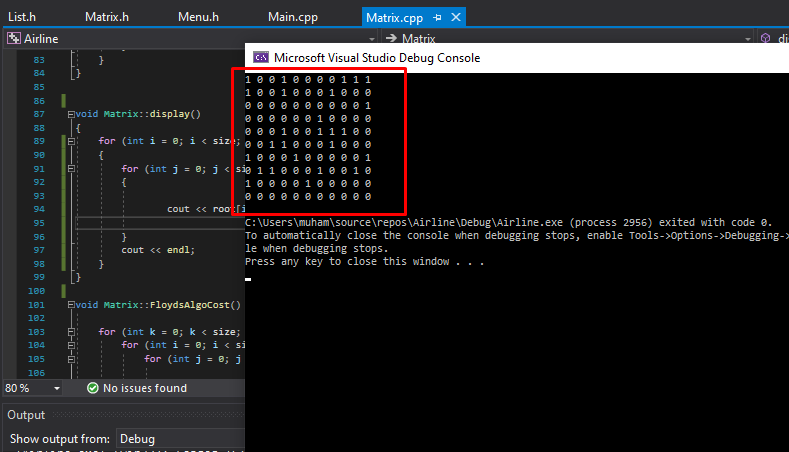
int hotelCharges;

bool isDestination;

**GRAPH:-**

We have used the **Node** to generate the **Adjacency matrix**. We have read the file line by line and where we have the origin->destination we have there set **isDestination=true.**

By implementing this we have the adjacency matrix.



It generates the complete perfect Adjacency matrix as we have verified manually. The first row is for the first city in the file and the columns associated with this row is destination from the city. It also stores minimum cost and time of the routes and path to reach the minimum routes according to time or cost.

Later, we have focus on the implementation of graph data structure. Based on given data, we have generated the Graph and then use different pattern and algorithms to find the best possible path to destination as per requirements. We created a list that stores all the flights from a given source to destination.

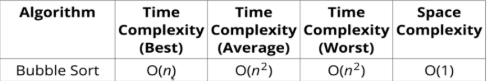
**LIST:**

We have implemented this class that stores data of all flights from one specific source to destination location using our implemented class of Route Characteristics. We have implemented it using Array Data Structure so that we can easily sort Flight Data on the basis of minimum cost and minimum time. The well sorted data after using Bubble Sort algorithm helps us with further proceeding of our project.

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Time Complexity**  **(Best)** | **Time Complexity**  **(Worst)** |
| List | O(1) | O(n) |

**Sorting:**

We have used the **Bubble Sort Technique** for the sorting of data Time and Cost.



We have sorted the data in the list class to make sure we have the minimum cost and time. We have used bubble sort because we have the small amount of data now.

In the list, we also compared/checked the user specified date and printed data accordingly, with correct date, airline(if any). Or if the date doesn’t match, then we showed alternatives.

**Floyd’s Algorithm:**

We have used the Floyd algorithm that generate shortest path on the base of minimum cost and one for minimum time according to the given scenarios.

The time complexity of this algorithm is:



**Working Program:**

