

PROJECT 4

ADJACENCY MATRIX GRAPH

CSCI 230
DATA STRUCTURE II

MAI PHAM

DEVELOPMENT ENVIRONMENT
MAC OS (xCode)

TABLE OF CONTENTS

- ❖ **Project Status**
- ❖ **Drawing of the Graph**
- ❖ **Annotated Test Cases**
- ❖ **Input/Output Results**
- ❖ **Source Code**

PROJECT STATUS:

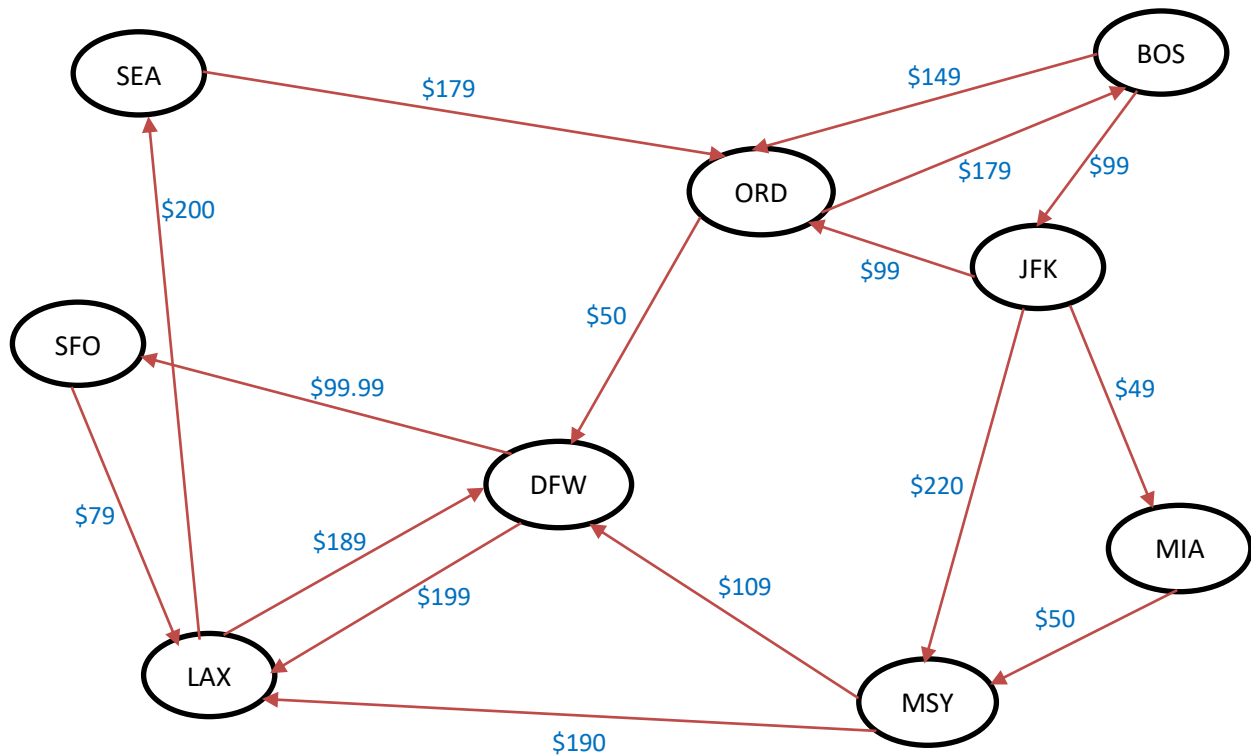
Objective

- ❖ Implement adjacency matrix for a weight graph regarding airports and airport's flights from two text files

Status

- ❖ For this project, I reused majority of the material from my previous labs. Then I just created additional functions/modified it to meet the project requirement. Overall, my project is completed and successfully run all the require selections and two extra credits (#6 and #9).

DRAWING OF THE GRAPH:



ANNOTATED TEST CASES:

There is one set of require test cases that was given for the project. It is the 1st set displaying in the output result below. It did verify majority of the project aspects like display all airport information (#0), display 1 airport information (#1), find cheapest flight from LAX to JFK and vice versa (#2 & #5), and add a flight between two airports (#3). Notice how at 1st, the flights from LAX to JFK actually cost \$657.5 with 4 stops. However, after add a flight from DFW to JFK, now the flight from LAX to JFK only cost \$389 with 2 stops. This have shown that my functions, add a flight and Dijkstra, are working correctly. This 1st set also included delete a flight (#4), nothing is change though since there is no route between the two input airports.

There are some selections that haven't been test in the 1st set, so I created a second set of test cases. In this set, I showed that the cheapest flights from JFK to LAX is \$289 with 3 stops (#2) but the fewest stop from JFK to LAX is 2 stops but cost \$410. I also retest delete a flight (#4) to show that my code is able to remove the flight from ORD to BOS. Then I added a new airport MSP and added some

departed and arrival flight for this new airport. All this information are showed at the end under selection 0 and in the two new output files.

INPUT/OUTPUT RESULTS:

Note: 1 set of test case as require from the project

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 0

Printing all airport information...

```
LAX -->      DFW($189)      SEA($200)
SF0 -->      LAX($79)
DFW -->      LAX($199)      SF0($99.99)
ORD -->      DFW($50)       BOS($179)
BOS -->      ORD($149)      JFK($99)
JFK -->      ORD($99)       MIA($49)      MSY($220)
MIA -->      MSY($50)
MSY -->      LAX($190)      DFW($109)
SEA -->      ORD($179.5)
```

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 1

Please enter the airport code = SF0

SF0 flight information...

```
SF0 -->      LAX($79)
SF0 <--      DFW($99.99)
```

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 2

Please enter the departure airport code = LAX

Please enter the arrival airport code = JFK

Cheapest Flight Route:

LAX --> SEA --> ORD --> BOS --> JFK

Total Stop = 4

Total Cost = \$657.5

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 2

Please enter the departure airport code = JFK

Please enter the arrival airport code = LAX

Cheapest Flight Route:

JFK --> MIA --> MSY --> LAX

Total Stop = 3

Total Cost = \$289

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 4

Please enter the departure airport code = LAX

Please enter the arrival airport code = SFO

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 3

Please enter the departure airport code = DFW

Please enter the arrival airport code = JFK

Please enter the total cost = 200

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 5

Please enter the departure airport code = LAX

Please enter the arrival airport code = JFK

Cheapest Flight Route:

LAX --> DFW --> JFK

Total Stop = 2

Total Cost = \$389

Cheapest Flight Route:

JFK --> MIA --> MSY --> LAX

Total Stop = 3

Total Cost = \$289

Round Trip Total Cost = \$678

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 0

Printing all airport information...

LAX -->	DFW(\$189)	SEA(\$200)	
SFO -->	LAX(\$79)		
DFW -->	LAX(\$199)	SFO(\$99.99)	JFK(\$200)
ORD -->	DFW(\$50)	BOS(\$179)	
BOS -->	ORD(\$149)	JFK(\$99)	
JFK -->	ORD(\$99)	MIA(\$49)	MSY(\$220)
MIA -->	MSY(\$50)		
MSY -->	LAX(\$190)	DFW(\$109)	
SEA -->	ORD(\$179.5)		

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> Q

Program ended with exit code: 0

```
P4FlightsRev1.txt
0      2      189.00
0      8      200.00
1      0      79.00
2      0      199.00
2      1      99.99
2      5      200.00
3      2      50.00
3      4      179.00
4      3      149.00
4      5      99.00
5      3      99.00
5      6      49.00
5      7      220.00
6      7      50.00
7      0      190.00
7      2      109.00
8      3      179.50
```

Note: Additional test cases set including two extra credits (option #6 & #9)

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 0

Printing all airport information...

```
LAX -->      DFW($189)      SEA($200)
SFO -->      LAX($79)
DFW -->      LAX($199)      SFO($99.99)
ORD -->      DFW($50)      BOS($179)
BOS -->      ORD($149)      JFK($99)
JFK -->      ORD($99)      MIA($49)      MSY($220)
MIA -->      MSY($50)
MSY -->      LAX($190)      DFW($109)
SEA -->      ORD($179.5)
```

Airport Flights Selection

1. Display airport information
2. Find a cheapest flight from one airport to another airport
3. Add a flight from one airport to another airport
4. Delete a flight from one airport to another airport
5. Find a cheapest roundtrip from one airport to another airport
6. Find a flight with fewest stops from one airport to another airport
7. Find all flight from one airport to another airport
8. Find an order to visit all airports starting from an airport
9. Add a new airport
- Q. Exit

Select an option ==> 2

Please enter the departure airport code = JFK

Please enter the arrival airport code = LAX

Cheapest Flight Route:

JFK --> MIA --> MSY --> LAX

Total Stop = 3

Total Cost = \$289

Airport Flights Selection

1. Display airport information
 2. Find a cheapest flight from one airport to another airport
 3. Add a flight from one airport to another airport
 4. Delete a flight from one airport to another airport
 5. Find a cheapest roundtrip from one airport to another airport
 6. Find a flight with fewest stops from one airport to another airport
 7. Find all flight from one airport to another airport
 8. Find an order to visit all airports starting from an airport
 9. Add a new airport
 - Q. Exit
- Select an option ==> 6

Please enter the departure airport code = JFK

Please enter the arrival airport code = LAX

Shortest Flight Route:

JFK --> MSY --> LAX

Total Stop = 2

Total Cost = \$410

Airport Flights Selection

1. Display airport information
 2. Find a cheapest flight from one airport to another airport
 3. Add a flight from one airport to another airport
 4. Delete a flight from one airport to another airport
 5. Find a cheapest roundtrip from one airport to another airport
 6. Find a flight with fewest stops from one airport to another airport
 7. Find all flight from one airport to another airport
 8. Find an order to visit all airports starting from an airport
 9. Add a new airport
 - Q. Exit
- Select an option ==> 4

Please enter the departure airport code = ORD

Please enter the arrival airport code = BOS

Airport Flights Selection

1. Display airport information
 2. Find a cheapest flight from one airport to another airport
 3. Add a flight from one airport to another airport
 4. Delete a flight from one airport to another airport
 5. Find a cheapest roundtrip from one airport to another airport
 6. Find a flight with fewest stops from one airport to another airport
 7. Find all flight from one airport to another airport
 8. Find an order to visit all airports starting from an airport
 9. Add a new airport
 - Q. Exit
- Select an option ==> 9

Please enter the airport code = MSP

Please enter the airport name = Minneapolis Saint Paul

Would you like to add flight(s) for the new airport (Y or N)? Y

Please enter the departure airport code = MSP
Please enter the arrival airport code = SEA
Please enter the total cost = 333

Add more flight(s) (Y or N)? Y
Please enter the departure airport code = SFO
Please enter the arrival airport code = MSP
Please enter the total cost = 223

Add more flight(s) (Y or N)? N

Airport Flights Selection

1. Display airport information
 2. Find a cheapest flight from one airport to another airport
 3. Add a flight from one airport to another airport
 4. Delete a flight from one airport to another airport
 5. Find a cheapest roundtrip from one airport to another airport
 6. Find a flight with fewest stops from one airport to another airport
 7. Find all flight from one airport to another airport
 8. Find an order to visit all airports starting from an airport
 9. Add a new airport
 - Q. Exit
- Select an option ==> 0

Printing all airport information...

LAX -->	DFW(\$189)	SEA(\$200)	
SFO -->	LAX(\$79)	MSP(\$223)	
DFW -->	LAX(\$199)	SFO(\$99.99)	
ORD -->	DFW(\$50)		
BOS -->	ORD(\$149)	JFK(\$99)	
JFK -->	ORD(\$99)	MIA(\$49)	MSY(\$220)
MIA -->	MSY(\$50)		
MSY -->	LAX(\$190)	DFW(\$109)	
SEA -->	ORD(\$179.5)		
MSP -->	SEA(\$333)		

Airport Flights Selection

1. Display airport information
 2. Find a cheapest flight from one airport to another airport
 3. Add a flight from one airport to another airport
 4. Delete a flight from one airport to another airport
 5. Find a cheapest roundtrip from one airport to another airport
 6. Find a flight with fewest stops from one airport to another airport
 7. Find all flight from one airport to another airport
 8. Find an order to visit all airports starting from an airport
 9. Add a new airport
 - Q. Exit
- Select an option ==> Q

Program ended with exit code: 0

P4AirportRev1.txt		
0	LAX	Los Angeles
1	SFO	San Francisco
2	DFW	Denver
3	ORD	Chicago
4	BOS	Boston
5	JFK	New York
6	MIA	Miami
7	MSY	New Orleans
8	SEA	Seattle
9	MSP	Minneapolis Saint Paul

P4FlightsRev1.txt		
0	2	189.00
0	8	200.00
1	0	79.00
1	9	223.00
2	0	199.00
2	1	99.99
3	2	50.00
4	3	149.00
4	5	99.00
5	3	99.00
5	6	49.00
5	7	220.00
6	7	50.00
7	0	190.00
7	2	109.00
8	3	179.50
9	8	333.00

SOURCE CODE:

Main.cpp

```
//  
// main.cpp  
// Project 4  
//  
// Created by Mai Pham on 5/26/18.  
// Copyright © 2018 Mai Pham. All rights reserved.  
//  
  
#include "AdjacencyMatrix.h"  
#include <iostream>  
using namespace std;  
  
int main() {  
    AdjacencyMatrix graph("P4Airports.txt", "P4Flights.txt");  
    char option;  
  
    cout << "Airport Flights Selection\n";
```

```
cout << "1. Display airport information\n";
cout << "2. Find a cheapest flight from one airport to another airport\n";
cout << "3. Add a flight from one airport to another airport\n";
cout << "4. Delete a flight from one airport to another airport\n";
cout << "5. Find a cheapest roundtrip from one airport to another airport\n";
cout << "6. Find a flight with fewest stops from one airport to another airport\n";
cout << "7. Find all flight from one airport to another airport\n";
cout << "8. Find an order to visit all airports starting from an airport\n";
cout << "9. Add a new airport\n";
cout << "Q. Exit\n";
cout << "Select an option ==> ";
cin >> option;
cout << endl;
while (option != 'Q') {
    string source, destination;
    double cost;
    switch (option) {
        case '0':
            cout << "Printing all airport information..." << endl;
            graph.printAllFlightInfor();
            cout << endl;
            break;
        case '1':
            cout << "Please enter the airport code = ";
            cin >> source;
            cout << source << " flight information..." << endl;
            graph.print1Flight(source);
            cout << endl;
            break;
        case '2':
            cout << "Please enter the departure airport code = ";
            cin >> source;
            cout << "Please enter the arrival airport code = ";
            cin >> destination;
            graph.cheapestFlight(source, destination);
            cout << endl;
            break;
        case '3':
            cout << "Please enter the departure airport code = ";
            cin >> source;
            cout << "Please enter the arrival airport code = ";
            cin >> destination;
            cout << "Please enter the total cost = ";
            cin >> cost;
            graph.addFlight(source, destination, cost);
            cout << endl;
            break;
        case '4':
            cout << "Please enter the departure airport code = ";
            cin >> source;
            cout << "Please enter the arrival airport code = ";
            cin >> destination;
            graph.deleteFlight(source, destination);
            cout << endl;
            break;
        case '5':
            cout << "Please enter the departure airport code = ";
            cin >> source;
            cout << "Please enter the arrival airport code = ";
            cin >> destination;
            double cost;
            cost = graph.cheapestFlight(source, destination);
            cost += graph.cheapestFlight(destination, source);
            cout << "Round Trip Total Cost = $" << cost << endl << endl;
            break;
        case '6':
            cout << "Please enter the departure airport code = ";
            cin >> source;
            cout << "Please enter the arrival airport code = ";
```

```

        cin >> destination;
        graph.shortestFlight(source, destination);
        cout << endl;
        break;
    case '7':
        cout << "Not Available" << endl;
        break;
    case '8':
        cout << "Not Available" << endl;
        break;
    case '9':
        char answer;
        string name, code;
        cout << "Please enter the airport code = ";
        cin >> code;
        cout << "Please enter the airport name = ";
        cin.ignore();
        getline(cin, name);
        graph.addNewAirport(code, name);
        cout << "\nWould you like to add flight(s) for the new airport (Y or N)? ";
        cin >> answer;
        while (answer == 'Y') {
            cout << "Please enter the departure airport code = ";
            cin >> source;
            cout << "Please enter the arrival airport code = ";
            cin >> destination;
            cout << "Please enter the total cost = ";
            cin >> cost;
            graph.addFlight(source, destination, cost);
            cout << "\nAdd more flight(s) (Y or N)? ";
            cin >> answer;
        }
        cout << endl;
        break;
    }
    cout << "Airport Flights Selection\n";
    cout << "1. Display airport information\n";
    cout << "2. Find a cheapest flight from one airport to another airport\n";
    cout << "3. Add a flight from one airport to another airport\n";
    cout << "4. Delete a flight from one airport to another airport\n";
    cout << "5. Find a cheapest roundtrip from one airport to another airport\n";
    cout << "6. Find a flight with fewest stops from one airport to another airport\n";
    cout << "7. Find all flight from one airport to another airport\n";
    cout << "8. Find an order to visit all airports starting from an airport\n";
    cout << "9. Add a new airport\n";
    cout << "0. Exit\n";
    cout << "Select an option ==> ";
    cin >> option;
    cout << endl;
}
graph.outputAirports();
graph.outputFlight();
return 0;
}

```

AdjacencyMatrix.cpp

```

//
// AdjacencyMatrix.h
// Project 4
//
// Created by Mai Pham on 5/26/18.
// Copyright © 2018 Mai Pham. All rights reserved.
//

#ifdef AdjacencyMatrix_h
#define AdjacencyMatrix_h

#include <vector>

```

```
//#include <stack>
//#include <queue>
#include <string>
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;

struct vertices {
    int index;           // not necessary
    string code;
    string name;
};

class AdjacencyMatrix {
private:
    double **adjMatrix;
    vertices vertArray [20];
    int numVertex;
public:
    AdjacencyMatrix(string airportFile, string flightsFile){
        int i, j;
        double m;
        string c, f;

        ifstream myAirport;
        myAirport.open(airportFile);
        if(!myAirport.is_open())
            cout << "No text file found. " << endl;
        while (myAirport >> i) {
            myAirport >> c;
            getline(myAirport, f);
            f = f.substr(4, f.length());
            //cout << i << "\n" << c << "\n" << f << endl;
            vertArray[i].index = i;
            vertArray[i].code = c;
            vertArray[i].name = f;
            //cout << vertArray[i].name << endl;
        }

        numVertex = i+1;
        //cout << numVertex << endl;
        adjMatrix = new double*[numVertex];
        for (int i = 0; i < numVertex; i++) {
            adjMatrix[i] = new double[numVertex];
            for (int j = 0; j < numVertex; j++)
                adjMatrix[i][j] = false;
        }

        ifstream myFlights;
        myFlights.open(flightsFile);
        if(!myFlights.is_open())
            cout << "No text file found. " << endl;
        while (!myFlights.eof()) {
            myFlights >> i >> j >> m;
            //cout << i << "\n" << j << "\n" << m << endl;
            addEdge(i, j, m);
        }
    }

    AdjacencyMatrix(int v) {
        numVertex = v;
        adjMatrix = new double*[numVertex];
        for (int i = 0; i <= numVertex; i++) {
            adjMatrix[i] = new double[numVertex];
            for (int j = 0; j <= numVertex; j++)
                adjMatrix[i][j] = false;
        }
    }
}
```

```

~AdjacencyMatrix() {
    for (int i = 0; i < numVertex; i++)
        delete[] adjMatrix[i];
    delete[] adjMatrix;
}

void addEdge(int i, int j, double w) {
    if (i == j)
        return;
    if (i >= 0 && i < numVertex && j >= 0 && j < numVertex)
        adjMatrix[i][j] = w;
}

void removeEdge(int i, int j) {
    if (i == j)
        return;
    if (i >= 0 && i < numVertex && j >= 0 && j < numVertex)
        adjMatrix[i][j] = false;
}

int returnIndex(string code) {
    int index = -1; // return -1 if not found
    for (int i = 0; i < numVertex; i++) {
        if (vertArray[i].code == code)
            index = i;
    }
    return index;
}

void addFlight(string source, string dest, double cost) {
    int s, d;
    s = returnIndex(source);
    d = returnIndex(dest);
    if (s != -1 && d != -1)
        addEdge(s, d, cost);
    else
        return;
}

void deleteFlight(string source, string dest) {
    int s, d;
    s = returnIndex(source);
    d = returnIndex(dest);
    if (s != -1 && d != -1)
        removeEdge(s, d);
    else
        return;
}

void printMatrix() {
    for(int i = 0; i < numVertex; i++) {
        for(int j = 0; j < numVertex; j++)
            cout << adjMatrix[i][j] << "\t";
        cout<<endl;
    }
}

void printAllFlightInfor() {
    for(int i = 0; i < numVertex; i++) {
        cout << vertArray[i].code << " -->";
        for(int j = 0; j < numVertex; j++)
            if (adjMatrix[i][j])
                cout << "\t" << vertArray[j].code << "($" << adjMatrix[i][j] << ")";
        cout<<endl;
    }
}

void print1Flight(string source) {

```

```

int src = returnIndex(source);
if (src != -1) {
    cout << vertArray[src].code << " -->";
    for(int j = 0; j < numVertex; j++)
        if (adjMatrix[src][j])
            cout << "\t" << vertArray[j].code << "($" << adjMatrix[src][j] << ")";
    cout<<endl;
    cout << vertArray[src].code << " <--";
    for(int j = 0; j < numVertex; j++)
        if (adjMatrix[j][src])
            cout << "\t" << vertArray[j].code << "($" << adjMatrix[j][src] << ")";
    cout<<endl;
}
else
    cout << "Found No Matching Airport" << endl;
}

double cheapestFlight(string source, string dest) {
    int src = returnIndex(source);
    int des = returnIndex(dest);
    int dist[numVertex];
    double cost[numVertex];
    bool sptSet[numVertex];
    int parent[numVertex];

    for (int i = 0; i < numVertex; i++) {
        parent[i] = -1;
        cost[i] = INT_MAX;
        sptSet[i] = false;
        dist[i] = 0;
    }

    cost[src] = 0;
    for (int count = 0; count < numVertex; count++) {
        int min = INT_MAX, min_index;
        for (int v = 0; v < numVertex; v++)
            if (sptSet[v] == false && cost[v] <= min) {
                min = cost[v];
                min_index = v;
            }
        int u = min_index;
        sptSet[u] = true;
        for (int v = 0; v < numVertex; v++)
            if (!sptSet[v] && adjMatrix[u][v] && cost[u] + adjMatrix[u][v] < cost[v]) {
                parent[v] = u;
                cost[v] = cost[u] + adjMatrix[u][v];
                dist[v] = dist[u] + 1;
            }
    }

    //for (int i = 0; i < numVertex; i++)
    //cout << parent[i] << endl;
    cout << "Cheapest Flight Route:" << endl;
    cout << " " << vertArray[src].code;
    printPath(parent, des);
    cout << "\n\tTotal Stop = " << dist[des];
    cout << "\n\tTotal Cost = $" << cost[des] << endl;
    /*
    for (int i = 0; i < numVertex; i++) {
        cout << src << " -> " << i << "\t\t" << dist[i] << "\t\t" << src << " ";
        printPath(parent, i);
        cout << endl;
    }
    */
    return cost[des];
}

void printPath(int parent[], int j) {
    if (parent[j] == -1)
        return;

```

```

    printPath(parent, parent[j]);
    cout << " --> " << vertArray[j].code;
}

double shortestFlight(string source, string dest) {
    int src = returnIndex(source);
    int des = returnIndex(dest);
    double cost[numVertex];
    int dist[numVertex];
    bool sptSet[numVertex];
    int parent[numVertex];

    for (int i = 0; i < numVertex; i++) {
        parent[i] = -1;
        dist[i] = INT_MAX;
        sptSet[i] = false;
        cost[i] = 0;
    }

    dist[src] = 0;
    for (int count = 0; count < numVertex; count++) {
        int min = INT_MAX, min_index;
        for (int v = 0; v < numVertex; v++)
            if (sptSet[v] == false && dist[v] <= min) {
                min = dist[v];
                min_index = v;
            }
        int u = min_index;
        sptSet[u] = true;
        for (int v = 0; v < numVertex; v++)
            if (!sptSet[v] && adjMatrix[u][v] && dist[u] < dist[v]) {
                parent[v] = u;
                dist[v] = dist[u] + 1;
                cost[v] = cost[u] + adjMatrix[u][v];
            }
    }

    //for (int i = 0; i < numVertex; i++)
    //cout << parent[i] << endl;
    cout << "Shortest Flight Route:" << endl;
    cout << " " << vertArray[src].code;
    printPath(parent, des);
    cout << "\n\tTotal Stop = " << dist[des];
    cout << "\n\tTotal Cost = $" << cost[des] << endl;
    /*
    for (int i = 0; i < numVertex; i++) {
        cout << src << " -> " << i << "\t\t" << dist[i] << "\t\t" << src << " ";
        printPath(parent, i);
        cout << endl;
    }
    */
    return dist[des];
}

void outputFlight(){
    ofstream outFile("P4FlightsRev1.txt");
    for(int i = 0; i < numVertex; i++) {
        for(int j = 0; j < numVertex; j++)
            if (adjMatrix[i][j])
                outFile << i << "\t" << j << "\t" << fixed << setprecision(2) <<
adjMatrix[i][j] << endl;
    }
}

void outputAirports(){
    ofstream outFile("P4AirportRev1.txt");
    for(int i = 0; i < numVertex; i++)
        outFile << i << "\t"<< vertArray[i].code << "\t\t" << vertArray[i].name << endl;
}

```

```
void addNewAirport(string code, string name){
    //cout << "current numVertex size " << numVertex << endl;
    vertArray[numVertex].code = code;
    vertArray[numVertex].name = name;
    numVertex++;
    //cout << "current numVertex size " << numVertex << endl;
    double **temp = new double*[numVertex];
    for (int i = 0; i < numVertex; i++) {
        temp[i] = new double[numVertex];
        for (int j = 0; j < numVertex; j++) {
            if (j == numVertex-1 || i == numVertex-1)
                temp[i][j] = false;
            else
                temp[i][j] = adjMatrix[i][j];
            //cout << "temp " << i << j << temp[i][j] << " ";
        }
    }
    delete [] adjMatrix;           //destroy everything in here
    adjMatrix = temp;              //reset with new size
}
};

#endif /* AdjacencyMatrix_h */
```