CSE3302/3020 - Software Engineering Project
Frequent Flyer Points Calculator

supervisor: Grace Rumantir
website: http://www.csse.monash.edu.au/~gwr/CSE3302
newsgroup: monash.csse.cse3302
email: gwr@csse.monash.edu.au

Motivation

Billabong Air, an Australian-based airline, has just formed an alliance with UKingdom Air and Yankee Air. This enables their frequent travellers to use their frequent flyer points to fly free on the networks serviced by the three airlines.

The few Frequent Flyer Points Calculators available on the internet are limited in their functionalities. Typically, they either support only major cities in one network or selected route between two cities with 1 transfer point. Customers would have to contact the airline to find out about the number of points required for more complicated routes. Billabong Air targets frequent corporate travellers whom most likely have a lot of points to redeem and would choose routes which involve a number of cities and airlines.

Dissatisfied with the packages that can be bought off-the-shelf, the Marketing Manager of Billabong Air approached a group of 6 enthusiastic third year students from CSSE - Monash University to come up with a prototype of their Frequent Flyer Points Calculator website.

The students see this as an opportunity to build a generic Frequent Flyer Points Calculator system which can be used by any airline and perhaps incorporates even more sophisticated functionalities beyond the requirements outlined by Billabong Air.

Understanding the Domain

The three airlines have their own frequent flyer programs. It has been decided that if a selected route is serviced by more than 1 airline, the total flight distance of each airline will be calculated and charged according to its individual program rule. The number of points required for the entire route is the sum of the number of points required by each airline. There is no restriction as to how many cities a traveller can visit except that he/she may not go via the city of origin. The Frequent Flyer Calculator should have the following basic functionalities:

1. Given two cities, calculate the minimum number of points needed to fly between them and output the corresponding route.
2. Given a number of redeemable points, find the potential shortest routes that a traveller can take. If an originating city is given, then output the routes starting from the originating city first with the rest of the routes sorted based on the number of points required.
3.

Given two cities and a maximum number of connections, find the possible routes a traveller can take.

4. Given a route, calculate the number of points earned or required

Each of the functionalities should produce a report using a template which shows the detail of each leg/connection of the trip, the names of the airports between them, the distance, the number of points required/earned with the respective airline. A summary of the total points required/earned (for a one-way or return trip) on individual airlines and a grand total should be provided at the end of the report.

Distance Between Two Airports

An airport has a world coordinate position expressed in latitude (South to North) and longitude (East to West) in degrees, minutes and seconds. Given the positions of two airports, L_1, N_1 and L_2, N_2, and the equatorial radius of the earth, R = 6378 kilometers, the distance (in radian) can be calculated using the approximated formulae called the Great Circle Distance:

$$Distance = \arccos[\sin L_1 * \sin L_2 + \cos L_1 * \cos L_2 * \cos(N_2 - N_1)] * \pi/180 * R$$

Note, before using the formulae, the unit of the positions of the two airports should be converted to positive radian first.

Redeeming Points

The award levels for Billabong Air are shown in the table below:

Zone	Max km	Points Required		
	Distance	Economy	Business	First
1	0-1,240	9,000	11,250	13,500
2	1,241-1,700	17,000	21,250	25,500
3	1,701-2,900	25,000	31,250	37,500
4	2,901-7,300	30,000	37,500	45,000
5	7,301-11,000	40,000	60,000	80,000
6	11,001-14,600	60,000	90,000	120,000
7	14,601-20,000	80,000	120,000	160,000
8	20,001-31,000	100,000	150,000	200,000
9	31,001-50,000	130,000	195,000	260,000
9	50,001-70,000	150,000	225,000	300,000

The award levels for UKingdom Air are shown in the table below:

Zone	Max km	Points Required		
	Distance	Economy	Business	First
1	0-965	25,750	51,500	77,250
2	966-1,610	32,250	64,500	96,750
3	1,611-2,255	35,500	71,000	106,500
4	2,256-3,220	40,250	80,500	120,750
5	3,221-6,440	48,250	96,500	144,750
6	6,441-8,050	56,250	112,500	168,750
7	8,051-12,880	64,500	129,000	193,500
8	12,881-19,965	80,500	161,000	241,000
9	19,966-28,980	112,750	225,500	338,250
10	over 28,980	128,750	257,500	386,250

Yankee Air Frequent Flyer program does not go by distance but charges a flat rate for a maximum of 5 connections shown below:

Zone	Max km	Points Required		
	Distance	Economy	Business	First
	flat rate	45,000	70,000	105,000

Earning Points

A Billabong Air Frequent Flyer member will earn points with the following formulae regardless of which airline he/she travels on:

Points per km	Fare
1.50	First Class
1.25	Business Class
1.00	Full Economy
0.70	Excursion/Discount Economy

Problem Solving

The project can be seen as an application of weighted undirected graphs covered in Data Structure and Algorithm subject. In graph terminology, cities/airports are *nodes* (or *vertices*), a direct flight between two airports is an *arc* (or *edge*) and the distance of a direct flight is the arc's *weight*. Since the connection between two airports is always two ways, the graph representing the network of an airline is said to be *undirected*.

Tasks

Each group is to come up with a structured analysis and design for the project. This has been covered in CSE3308 Software Engineering: Analysis and Design.

The first three functionalities are to be implemented using both adjacency matrix and adjacency list. The first module can be implemented using Djikstra's shortest path algorithm, the second using Floyd's all shortest paths algorithm and the third using Breadth First or Depth First Search algorithm. All modules are to be implemented in C.

The user interface is to be implemented in C-CGI program. Each module should have its own screen linked from the main menu. All of the modules should have user interface with consistent look and feel.

Students will use the frequent flyer programs mentioned above as the basis to build their generic system. However, to achieve good marks, students should try to introduce extensions to the basic functionalities.

It is up to the students to come up with a network for each of the three airlines to highlight the functionalities of their system. A complete list of airports in the world with the corresponding unique airport codes and world coordinate positions can be found at the website: http://www.calle.com/aviation/ (follow the link to: List Airports by Country and Data Quality).

We will use ICAO airport code system. For example, the code CYYZ can be dissected into C as the country code for Canada and YYZ as the airport code for Toronto Lester B Pearson International Airport. At the website, click on the airport name to get the exact latitute and longitude coordinate, e.g. Toronto Int'l Airport (code CYYZ) is at 43° 40' 37.999N latitude and 79° 37' 50.002W longitude.

All modules should use the same files. To build a network, two types of file need to be set up:

1. A master file. Each record comprises a unique airport code, the name of the airport, the city in which it is located and its location in latitude and longitude coordinate
2. A connection file. Each record represents a direct link between an originating and a destination airport. Each record in this file comprises two airport codes with the distance between them (calculated using the Great Circle Distance formulae defined above).

Division of Tasks

It is important that all group members get involved in the analysis and design of the project. The question of system modularity, cohesion and coupling should be addressed at this stage and reflected in the design of the common files and data structures.

At the minimum, each group member gets the task of implementing one algorithm for any of the functionalities using either the adjacency matrix or list as explained in the previous section. The documentation of each module should include test cases to show that the module functions properly.

The implementations of the common modules, templates for the systems's user interface and report to ensure consistent look and feel and modularity are to be shared evenly. The basic common modules include the distance calculator, function to encode/decode the airport name/code, functions to build a network (using both the adjacency matrix and list) and function to calculate the number points earned/required for a given route.

For obvious ethical reason, all material/code downloaded from the web which are used for the project should be properly cited.

Assessment

The project is assessed 40% based on the quality of the group work and 60% on the individual work. All group members will get equal mark for the group work if and only if each individual member puts in fair contribution toward the work.

Two forms will be provided for group members to fill in. The first form is to outline which part of the project each student is responsible for. This form should be signed by all members of the group. The second form is to be submitted by each individual group member who will independently judge the level of contribution of each member of the group. The forms should be submitted together with the final report.

At the end of the semester, there will be a demonstration session in which each student will answer questions on his/her part of the project. Each group should submit a single user guide for the project. Programmer documentation should be supplied for all source module implementation. This documentation functions as a programmer's guide. The breakdown of the mark assigned to each student is shown in the table below:

Table: Marking scheme

Submission	Assessment	Due
Group: Project analysis and design	25%	6/9/99
Individual: student's code, technical documentation and user guide	60%	22/10/99
Group: integration of system	15%	22/10/99
Group: project demonstration in FCIT 208 at 9.00 am	compulsory	22/10/99

Group Formation, Meetings and Computer Access

Groups of six students have been formed and can be seen on the website http://www.cs.monash.edu.au/~allocate/. Students can still move between groups until 30/7/99.

The first meeting will be on Friday, 9/8/99 at 11.00 am in room 135 to discuss the overall scope of the project, finalize the group formation and arrangements for the weekly meeting between the supervisor and each individual group. The weekly meeting is on Fridays in room 135 to discuss milestones of the project in general. Two groups will have the same one-hour weekly meeting on the hour. For example, groups with time slots 9.00-9.30 and 9.30-10.00 will have the weekly meeting at 9.00. All group members should attend the weekly meeting. Each project group should arrange group meetings on their own to discuss the design of their system.

Each group should setup a project webpage. CSE3302 website has a link to the procedure of how to set up a website on Silas. Lab 139, 207 and 208 in the FCIT building have been booked throughout the semester for third year students to work on their projects unsupervised.

Important information for the project will be posted on the project website and the third year notice board. Discussions of general interest related to the project will be conducted on the project's newsgroup: monash.csse.cse3302.