



HACETTEPE UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
BBM204
PROGRAMMING ASSIGNMENT #3

Subject : Graphs
Submission Date : May 2, 2013
Deadline : May 12, 2013
Programming Language: ANSI C (89)
Advisors : R.A. Cemil ZALLUHOĞLU

Problem

In this experiment, you will practice about undirected graphs and you will implement a traveler problem. Imagine that you are a traveler and you want to go from a city to another city. In this problem, you have several choices for the transportation type. There may not be a direct path from you the source to the destination city, and thus you have to pass some other cities to reach the destination. Passing through the cities, you need to choose the transportation type of the path connecting these cities.

Transportation Types

- Highway
- Airway
- Railway



Program Operations

Part I

You will get the input file named `transportation_network.inp` that includes the adjacency matrixes of cities for three transportation types:

```
Highway
Cities City1 City2 ... CityN
City1 0/1 0/1 ... 0/1
City2 0/1 0/1 ... 0/1
.
.
CityN 0/1 0/1 ... 0/1
Airway
Cities City1 City2 ... CityN
City1 0/1 0/1 ... 0/1
City2 0/1 0/1 ... 0/1
.
.
CityN 0/1 0/1 ... 0/1
Railway
Cities City1 City2 ... CityN
City1 0/1 0/1 ... 0/1
City2 0/1 0/1 ... 0/1
.
.
CityN 0/1 0/1 ... 0/1
```

A sample matrix for the Airway is shown below. **Please note in your code matrix size should be dynamic. Static arrays will be penalized.**

	Hava Yolu									
Şehirler	İstanbul	İzmir	Ankara	Antalya	Kayseri	Hatay	Trabzon	Şanlıurfa	Kars	Hakkari
İstanbul	0	1	1	1	0	0	0	0	1	0
İzmir	1	0	0	0	0	1	0	0	0	1
Ankara	1	0	0	1	0	0	0	0	0	0
Antalya	1	0	1	0	1	1	1	0	0	0
Kayseri	0	0	0	1	0	1	0	1	0	0
Hatay	0	1	0	1	1	0	0	1	1	0
Trabzon	0	0	0	0	0	0	0	1	1	0
Şanlıurfa	0	0	0	0	1	1	1	0	0	1
Kars	1	0	0	0	0	1	1	0	0	1
Hakkari	0	1	0	0	0	0	0	1	1	0

In addition to the matrices in the input file, you need to read another file named `query.inp` which includes a number of queries about possible travels such as which transportation types and how many transportations types you are asked to use while you are going from one city to another. In this experiment you have got a three different query type:

- `Q1 City1 City2 {Hi}0/1{Aj}0/1{Rk}0/1` where H denotes Highway, A denotes Airway, R denotes Railway, and *i, j, k* are some integers.

For this query, you need to print the path satisfying the provided conditions, if there is any such path. You must find the path **according to the order in the query** and **it is enough to show one right path**.

For example, `Q1 Istanbul Kars A3 R2 H1` means the source city is Istanbul; the destination is Kars, and you must use 3 airway, 2 railway, 1 highway to reach Kars. A sample output could be:

```
Istanbul A Ankara A Kayseri A Sanliurfa R Hakkari R Trabzon H
Kars
```

- `Q2 City1 City2 N` with N denoting an integer.

For this query, you need to list the paths from `City1` to `City2` by passing through N different cities, if any such path exists. **Your code must return all possible paths.**

For instance, `Q2 Hatay Kayseri 3` means the source city is Hatay, the destination is Kayseri, and you are required to list all possible paths while you are going from Hatay to Kayseri by passing through 3 cities. A sample output could be:



Q2 Hatay Kayseri 3

```
Hatay R Antalya H İzmir R Ankara H Kayseri
Hatay R Antalya H İstanbul H Ankara H Kayseri
Hatay H Ankara H İzmir R Antalya H Kayseri
Hatay H Ankara H İzmir R Antalya H Kayseri
...
```

- Q3 City1 City2 Type where Type can be H for Highway, A for Airway R for Railway.

For this query, you need to list the paths from City1 to City2 by going over a specific transportation type, if any such path exists. **Your code must return all possible paths.**

For example, Q3 Antalya Hakkari R means the source city is Antalya, the destination is Hakkari, and you are required to list all possible paths while you are going from Antalya to Hakkari by using only railways. Please note that Hatay A Ankara and Hatay R Ankara are two different paths. A sample output could be:

```
Antalya Hatay Sanliurfa Hakkari
Antalya Hatay Sanliurfa Trabzon Kars Hakkari
Antalya Hatay Kayseri Sanliurfa Hakkari
Antalya Hatay Kayseri Sanliurfa Trabzon Kars Hakkari
```

Bonus Part 1

If the user enter Q1 type query Q1, you must find a path by not considering the given order of transportation types but only the number.

Ex:

```
Q1 Istanbul Kars A3 R2 H1
```

In Normal Part you must find a path according to given transportation order shown below.

```
Istanbul A Ankara A Kayseri A Sanliurfa R Hakkari R Trabzon H Kars
```



In Bonus Part you may find a path such as

Istanbul **R** Ankara **A** Kayseri **H** Sanliurfa **A** Hakkari **R** Trabzon **A** Kars

Bonus Part 2

You must add a new city in your graph in the query file. After you add the city, you must add a new path to reach that city. **(Note: You cannot add a path unless you add a new city)**

Command:

```
ADD City Bursa
ADD Path Bursa Istanbul H
ADD Path Bursa Hatay A
...
```

Evaluation

- You have to get everything (Transportation matrices, Queries, Adding process command from the input files.
- When your code evaluated, one input file will be given to your code and will be expected an output file that consists the result of your queries is expected.
- In your Report you must draw and explain your graph. You have to combine all transportation types in one graph. Because of that, you must use **colors** for you graph. **(Note: Colorless graphs will be penalized)**

Submit Format

```
Exp3.zip/ (Required)
report/; (Required)
report/*.pdf; (Required)
src/;(Required)
src/Makefile; (Required)
src/main.c; (Required)
src/*.c; (Optional)
src/*.h; (Optional)
```



NOTES AND RESTRICTIONS:

- Your experiment should be submitted before the due date. Late submissions will be penalized.
- All assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudo code) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.