PROGRAMMING ASSIGNMENT 4

Subject : Directed Graphs, Graph Search
Date Due : 21.04.2016
Programming Language : Java
Advisors : Dr. Adnan ÖZSOY, Dr. Erkut ERDEM, Dr. Gönenç ERCAN, R.A. Cumhur Yiğit ÖZCAN

1. INTRODUCTION

Graphs are data structures used to model pairwise relations between objects. A graph consists of vertices that are used to model the objects and edges connecting two vertices that are used to model the pairwise relation between those objects. Common usages of graphs include problems such as finding shortest path between two points, coloring state maps etc.

2. EXPERIMENT

2.1. PROBLEM

You are expected to develop a flight search engine that enables users to make flight plans. The flight database will be provided via two input files. Your program should read these input files and create a directed graph structure to store the information.

2.1.1 Airport List

The first input file contains a list of cities along with aliases of the airports located at those cities. Each line of this file contains the name of a city and the airport aliases which are separated by *tab* character. The city names and airport aliases will be unique among the system.

<***city_name***><tab><***airport_alias***>[<tab><***airport_alias***>]

2.1.2 Flight List

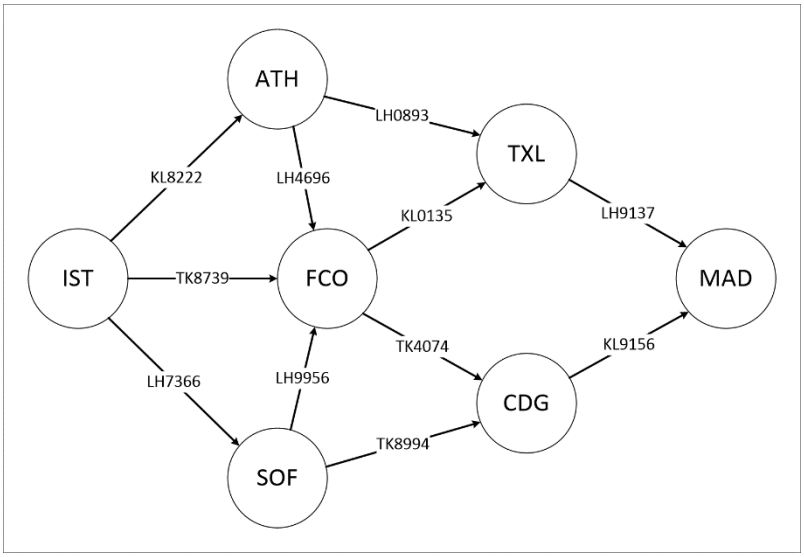
The second input file contains a list of all the available flights to be included in the database. Each line represents an individual flight containing the flight id, departure and arrival airport aliases, the departure date and time (as hours and 3minutes), flight duration (again, as hours and minutes) and the price respectively.

<***flight_id***><tab><***dept***>-><***arr***><tab><***dept_date***><tab><***duration***><tab><***price***>

The flight ids are composed of two uppercase characters that represent one of the airlines companies that are:

- Turkish Airlines (TK)
- Royal Dutch Airlines (KL)
- Lufthansa Airlines (LH)
- Delta Airlines (DL)
- American Airlines (AA)

and a four digit number to ensure that the flight ids are unique among the system.



2.2. COMMANDS

After reading the two input files mentioned above, your program should read from a third input file that contains the commands which are essentially flight search queries. A flight search query yields a flight plan which is either a single flight or a series of connected flights from the departure point to the arrival point. There are two rules for two flights to be eligible to be connected in a flight plan:

- Arrival point of the first flight and departure point of the second flight should be the same *airport*.
- Departure time of the second flight should be later than arrival time of the first flight.

There are also two more points that should be taken into consideration while forming a flight plan that are:

- Transferring between two airports at the same city is **not** allowed.
- A flight plan may **not** pass through the same city more than once. This rule applies to all the airports at a city.

Below is a list of 8 different commands. Each command includes a departure point and an arrival point that are *cities*. The start date parameter of the commands indicates the earliest date (starting from midnight, 00:00) the flights may start and it is possible for a flight plan to start any time after the start date.

2.2.1 List All

List all command is used for listing all the possible flight plan(s) from departure point to the arrival point.

Command format:
listall<tab><***dept***>-><***arr***><tab><***start_date***>

2.2.2 List Proper

List proper command is used for listing all the proper flight plan(s) from departure point to the arrival point which means a flight plan should be removed from the resulting flight plan set if there is another flight plan which is both quicker (that arrives at the final destination earlier) and cheaper.

Command format:
listproper<tab><***dept***>-><***arr***><tab><***start_date***>

2.2.3 List Cheapest
List cheapest command is used for listing the cheapest possible flight plan(s) from departure point to the arrival point. Keep in mind that there may be more than one possible cheapest flight plans.

Command format:
listcheapest<tab><***dept***>-><***arr***><tab><***start_date***>

2.2.4 List Quickest
List quickest command is used for listing the quickest possible flight plan(s) from departure point to the arrival point. Keep in mind that there may be more than one possible quickest flight plans.

Command format:
listquickest<tab><***dept***>-><***arr***><tab><***start_date***>

2.2.5 List Cheaper
List cheaper command is used for listing all the ***proper*** flight plans from departure point to the arrival point that are cheaper than given parameter.

Command format:
listcheaper<tab><***dept***>-><***arr***><tab><***start_date***><tab><***max_price***>

2.2.6 List Quicker
List quicker command is used for listing all the ***proper*** flight plans from departure point to the arrival point that arrive to the destination earlier than given parameter.

Command format:
listquicker<tab><***dept***>-><***arr***><tab><***start_date***><tab><***latest_date_time***>

2.2.7 List Excluding
List excluding command is used for listing all the ***proper*** flight plans from departure point to the arrival point that do not involve a flight from given airlines company.

Command format:
listexcluding<tab><***dept***>-><***arr***><tab><***start_date***><tab><***company***>

2.2.8 List Only From
List excluding command is used for listing all the ***proper*** flight plans from departure point to the arrival point that consists of flights only from the given airlines company.

Command format:

listonlyfrom<tab><***dept***>-><***arr***><tab><***start_date_time***><tab><***company***>

2.3. Output

Results of each command is a list of flight plans that should be printed to an output file. Simply, each flight plan should be printed to a line of the output file in the following format:

<***flight_id***><tab><***dept***>-><***arr***>[||<***flight_id***><tab><***dept***>-><***arr***>]<tab><**duration**>/<**total_price**>

If the search query cannot find any flight plan to list, the output should be "No suitable flight plan is found". For a detailed information about the output file please check sample I/O set.

4. LAST REMARKS

- Input files are going to be **syntactically and semantically error free**. You are **not** expected to check for any erroneous entry.
- The file that contains the main method should be named "Main.java".
- You are free to use any pre-defined class/package from JDK 1.8.0.
- The input and output file paths are **not** constant. They will be given as program arguments from the command line. Argument ordering will be as follows:
  <airport_list_file> <flight_list_file> <commands_file> <output_file>
- Sample input and output files can be found on ftp server. Examine them carefully before you start coding and ask any question you have via communication channels mentioned at the *notes and restrictions* section.

5. NOTES and RESTRICTIONS

- The output of your program will be graded automatically. Therefore, any difference of the output (even a smallest difference) from the sample output will cause an error and you will get 0 points from execution. **Keep in mind that a program that does not work %100 right is a program that works wrong.**
- Your work should obey general software engineering principles and conventions. (Design, Comments, Indentation)

  http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html
  http://java.sun.com/j2se/javadoc/writingdoccomments/

- You will use online submission system to submit your experiments. https://submit.cs.hacettepe.edu.tr/ (No other submission method such as e-mail will be accepted.)
- Final Submission Format:

```
<student_number>.zip
        |---- src
                |---- Main.java
                |---- *.java
```

- **SAVE** all your work until the assignment is graded.
- The assignment must be original, **INDIVIDUAL** work. Duplicate or very similar assignments are both going to be punished.
- You can ask your questions through course's piazza page (**BBM204**). **You are supposed to be aware of everything discussed in the page.**
- The advisor's office hours are Monday, 14:30 – 17:00 (R.A. Cumhur Yiğit ÖZCAN)