CS 150: Project 3 -

C W Liew

Version as of: 16:22 Sunday 19th November, 2017
Due: 11:55pm, Saturday, December 9, 2017

1 Introduction

2 Project Description

We are now somewhat in the world of the future - think **BladeRunner**. Humans no longer have to walk on the streets. Instead the streets have been converted to conveyor belts that carry you from one end to another. For simplicity and cost reasons, the conveyor belts are all uni-directional (one way) and therefore so are the streets. The city we are in has housing accomodations (hotels etc) and tourist sites (examples of interest from the 20th century - car factories, television stations etc). You are tasked with designing a program that will provide a suggested tourist route that will take a place from their starting point (a hotel) and take them to as many tourist sites as possible within an alloted amount of time.

3 Program Behavior

The program will read in two files: (1) a map of the city (directed graph) that has node labels (names) and edge weights; (2) a list of node labels, type (hotel or tourist site) and the suggested amount of time to be spent (if a tourist site). The program will buid its graph representation and then respond to queries from the user. Each query takes the form of: (1) starting node label, (2) desired time, and (3) a list of desired sites to include. For example:

```
51 500 60 93
```

This is a request for a tour starting (and ending) at node 51 that takes less than 500 minutes and includes visits to nodes 60 and 93. The program will print out the generated tour (including all the intermediate nodes) and the expected total time.

All user interactions will be through the console.

4 Program Input

The program inputs are:

1. city map (graph) in a file called ”city.txt”

2. hotels and tourist sites in a file called ”sites.txt”

Project Constraints

The following constraints apply to the project:

1. The project is to be completed individually. The only person you can consult is the instructor.
2. You are only to use containers from the Java Collections Framework. The only exception is when you use a method from an API that requires the use of arrays (e.g., main()).

5 Simplifications

You can simplify your project in one (or more) of the following ways:

- the user cannot specify the desired sites

6 Report

Your report should try to answer questions (you should design your own questions) like the following:

7 Submission

The project submission is in 4 parts:

1. (2.5 pts) - due Saturday, Dec 2 : the ”story” of your program
2. (2.5 pts) - due Monday, Dec 4: class diagrams for your program
3. (85 pts) - due Saturday December 9 : the rest including a draft report
4. (10 pts) - due Friday, December 15: the final report

s

Your submission for Part 3 will be composed of the following:

1. source files (*.java) that are commented and have javadoc directives
2. test files, one test file per class
3. a README.txt file that contains instructions on how to run your program
4. a draft of the project report (see project report guidelines `www.cs.lafayette.edu/~liew/courses/cs150/writeup-guidelines.pdf`) - 50% of the report grade is assigned to this. The remaining 50% will be given to a final report. The final report will be due several days after the draft is corrected and returned.