CSCE 3110 - Data Structures and Algorithms

Fall 2022

Course Project: World Airline Route Search

Due date and grade:

- Due date: Dec. 9, 2022 (No late submissions are accepted!)
- Full points: 100 (report: 40 points, code: 60 points)

Introduction:

In this project, you will be designing and implementing algorithms to store and search graphs. World Airline (WA) flies to many destinations worldwide including: Moscow, Seoul, Tokyo, Hong Kong, and London. Complete detailed information regarding airline flight routes can be found from the link flight.txt. The flight.txt contains a "from city" and its destination cities that WA flights to. Note that the flight.txt is a synthetically generated datasets by a graph data generator available from the link graphGen. You may want to download the graph generator and generate a few more graphs with different number of cities to test your algorithms.

Your job:

Your task is to design and implement algorithms to answer the following questions:

1. I am in city "A", can I fly to city "B" with less than x connections? Give me the route with the smallest number of connections or tell me there is no such a route.
2. Give me the route with the smallest number of connections from city "A" to city "D" through city "B" and "C". (the order of "B" and "C" is not important). Or tell me there is no such a route.
3. I am in city "A", my friend John is in a different city "B", and my other friend Ann is in yet another different city "C". We want to find a city different from the three cities we are in to meet so that the total number of connections among three of us is minimized. Tell me the city we should fly to and the routes for us or tell me there is no such a city.

You will need to process the documents and store their content in the data structures that you selected. Next, for every question, you will search your data structure using an algorithm of your choice. You have the freedom to select the data structures and algorithms that you consider to be more efficient for the tasks. Of course, you will have to justify your decisions. No fancy user interface is expected (fancy interface may be counted towards extra credits though). You program should take a few parameters with the first as the number of the question (1-4). The rest of the parameters and output of the four questions (1-4) are described as follows assuming your program is called routeSearch:

1. routeSearch 1 <city_A> <city_B> <num_connection>

   sample output:

      city_A to city_x to city_y … to city_B

      total connection: 4

2.

   usage: routeSearch 2 <city_A> through <city_B> and <city_C> to <city_D>

   sample output:

      city_A to city_x to city_C to city_y … to city_B to city_D

      smallest number of connection: 4

3.

   usage: routeSearch 3 <city_A> <city_B> <city_C>

   sample output:

      You three should meet at city_D

      Route for first person: city_A to city_x … to city_D (3 connections)

         Route for second person: city_B to city_y … to city_D (1 connections)

         Route for second person: city_C to city_y … to city_D (0 connections)

         Total number of connection: 4

Note: if it takes too long for your program to finish the tasks, try to generate smaller graphs using the provided graphGen.cpp.

What to turn in:

There are two main parts in this project, all of them contributing to the final project grade.

1. You will have to write a project report (about 3-5 typed pages – single space – 12pt font) that includes:
   - design issues, what are the problems and how you solve them
   - data structures you use, the decision behind selecting them
   - algorithms you employ, again with a justification of your decision
   - particular emphasis should be placed on the running time of your algorithm, please show the asymptotic costs for each of your algorithm
   - optimization issues: what could you do to further optimize your algorithm
   - you need to specifically address the problem of scalability: would you implementation be efficient in the case of very large species collections for larger scale geographic area?
   - any other remarks about your design and implementation

2. You will have to send in a fully working program, written in C/C++. We will test your algorithms extensively by generating graphs with different characteristics. It is mandatory that you include a README file, as detailed as possible, including compilation and running instructions. Is it also mandatory that your programs are fully documented, that is they should include detailed comments on what is included in each file and what each method does.

Submission: Submit a PDF report and a single zip file for your code to Canvas.

Notes:

No late submissions are accepted!.