EE 441
PROGRAMMING ASSIGNMENT 2

Trees & Graphs

Due Date: 11.12.2020 at 23.55
Form of Delivery: You should upload a single zip filewith your solution on ODTUClass. The filename should be *EE441_PA2_firstname_lastname_studentID.zip*. Also indicate how much time you spent for the homework (see Regulations).

Important Notice— The following programming assignment requires theoretical knowledge on binary search trees, bidirectional graphs and further methods on these data structures. Should you have a lack of experience on the subject, it is highly recommend that lecture notes are revisited beforehand.

EE Airlines

EE Airlines is a newly established local air travel company that operates between the airports of Republic of Eeland. The employees require certain software to conduct tasks of recording flight routes, searching for them, establishing new ones and discarding those that are unprofitable. Besides, additional software is required for a 3rd parties, namely airport staff and travel agencies for scheduling, pricing, canceling of flights etc.

a) The flight route map of EE airlines consists of the following routes illustrated in the map given below. For their very first month of the market, the initial plan, along with the route fares for the corresponding routes, needs to be archived.

Hint: *Use the bi-directional graph data structure for this purpose and & implement the global function* INITROUTEMAP. *City names in Eeland are provided in the* **citylist.txt** *file, which you can read into an array using the code given in the Appendix. Route fares for each route can be determined using a random function.*
Preferably, it is be beneficial to implement the public methods of ADDGCITY, REMOVEGCITY, SEARCHGCITY, ADDGROUTE, REMOVEGROUTE, SEARCHGROUTE, LISTGROUTES *to use in this and upcoming tasks.*

b) The monthly flight plan, derived from the flight route map created, is to be listed in order to allow the airport staff to use in boarding activities. Flight numbers are determined as 6-character strings, EEWXYZ first two of which is fixed as the company initials. WX is the numerical equivalent of the code of the city of departure, YZ is that of the city of arrival.

```
Ex: EE1506 is a flight from Signale (City 15) to East Circuitta (City 6).
```

Hint: *Store the flights in a binary search tree implement & the global function* INITFIGHTPLAN *that builds up the flight plan based on the route map.*
Note that implementing public ADDTFLIGHT, REMOVETFLIGHT, SEARCHTFLIGHT *methods may help in this and upcoming tasks.*

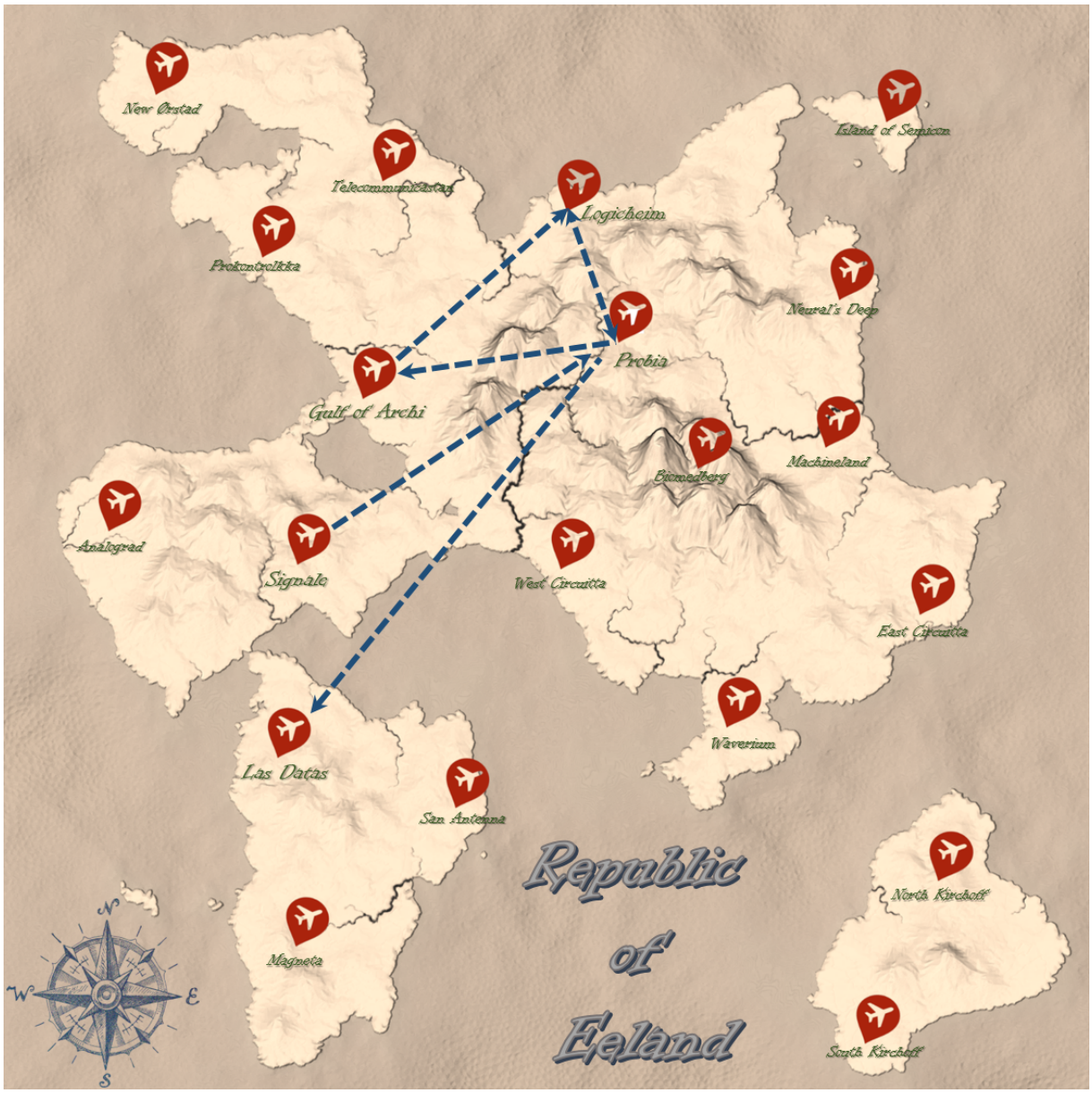

Fig. 1. Map of Republic of Eeland

c) The company offers advertises N new routes starting form the beginning of each month each month; hence, these need to be added first to the existing route map, then to the existing flight plan.
Hint: *Add the global function,* ADDMONTHLYNEWROUTES *in which you update the routemap and print the newly added route information. Then, implement* ADDMONTHLYNEWFLIGHTS*, another global function that adds the monthly new flights in the same order as the routes are added and prints the flight information to indicate addition.*
You may determine new routes by randomly selecting the place of departure and place of arrival from the list provided to you & determine whether the random selections are already existing or not. City names in Eeland are provided in the **citylist.txt** *file, which you can read into an array using the code given in the Appendix.*

d) The travel agencies keep the passenger occupancy rates as feedback to airway companies. Using this information, at the beginning of each month EE Airlines halves the ticket prices for flights with low occupancy rate (lower than OT%) in the preceding month and cancels flights with low occupancy rate in consecutive 2 months.Then, these routes are removed from the company's route map.

Hint: *Implement global functions* CANCELUNPOPULARFLIGHTS *&* DISCARDUNPOPULARROUTESANDUPDATEPRICES *to update flight fares, remove unprofitable flights and routes, consecutively. These functions are also expected to print information corresponding data of removal.*

You may add a new private field, OCCUPANCYSTATE *to the tree class, and the public methods* SET/GETOCCUPANYSTATE *to check and set whether the flights are popular among customers, (a random function will be more than enough to be used inside the "set" method to calculate monthly values). Ticket prices correspond to the route fares on the graph. You may implement public methods* SET/GETTICKETPRICES.

e) The route map and the monthly flight plan needs to be checked by the planners, every 6 months, such that each route has a return connection and each flight has a return flight. If not, those lacking return need to be added and advertised.
Hint: *You may implement global functions* CHECKNADDRETURNSROUTES *&* ADDRETURNFLIGHTS. *so that first routes are established, then flights are arranged accordingly, and corresponding information is printed.*

f) The travel agencies satisfy the customer demands by offering them different flight options, namely, direct, 1-stop, 2-stop flights.
Hint: *Add a global function that inputs the route map and the flight plan to return the flight options(if applicable) and their corresponding flight and price information when place of destination and arrival are entered.*

g) EE Airways publishes its monthly route map & flight report after all the updates. Provide a semi-annual (6-month) report.
Hint: *The report will automatically generate with the above implemented functions when the timing constraints are correctly applied in the main function. N=5, as the monthly new route and OT=70 as the occupancy threshold can be used as exemplar values.*

Regulations

- You should insert comments to your source code at appropriate places without including any unnecessary detail. **Comments WILL be graded.**
- Submit the whole Code::Blocks project folder in a zip file. The file name should be *EE441_PA1_firstname_lastname_studentID.zip*
- Indicate how much time you spent for the homework in a comment at the top of your main.cpp.
- Late submissions are welcome, but penalized according to the following policy:
  - 1 day late submission: HW will be evaluated out of 70 points
  - 2 days late submission: HW will be evaluated out of 50 points
  - 3 days late submission: HW will be evaluated out of 30 points
  - 4 or more days late submission: HW will not be evaluated

APPENDIX

The code to extract city names from the citylist.txt into a string array is provided below.

```
#include <fstream>
#include <string>
using namespace std;

int main () {
  string line;
  string city[20];
  ifstream myfile ("citylist.txt");
  if (myfile.is_open())
  {
   int i=0;
    while ( getline (myfile,line) )
    {
      city[i]=line;
      cout << city[i] << '\n';
      i++;
    }
    myfile.close();
  }

  else cout << "Unable to open file";

  return 0;
  }
```