Project 2

Air Travel Flight Management System

Air transport plays a crucial role in global connectivity, triggering significant impacts on mobility, trade and cultural interactions. The physical distance between countries and continents is thus shortened by the efficient interconnection provided today by the airlines around the world. The ability to plan and understand the complex flight network is extremely relevant for travelers and companies, so that decisions about travel and associated investments are as informed and efficient as possible.

In this context, the implementation of a system capable of facilitating access and understanding of the air transport network becomes an important tool. The main objective of this project is precisely to develop a flight management system for the air travel network of the airlines around the world that provides effective assistance to users who wish to explore and plan travel. The developed system is expected to meet the functional requirements, described in detail in this document, and to stand out for its usability and ability in providing relevant information in an intuitive and accessible manner.

To achieve this purpose, real data will be available that covers information about existing airports, including their geographic location, details about available flights, specifying departure and arrival airports, as well as the operating airline. A comprehensive list of airlines is also available. In the **dataset.zip** file, you can find all this information about a real air travel network. A detailed description of the provided dataset is given below and the following figure was obtained from this dataset, illustrating the position of the airports and existing flights between them.

Dataset

The dataset is available in **dataset.zip** in **csv** (*comma separated values*) format, as described below.

- File **airports.csv**: contains the identification information of *3019 airports*.

  The first line includes the headers: *Code* (unique IATA code of airport), *Name* (name of airport), *City* (city where the airport is), *Country* (the country), *Latitude* (latitude where the airport is), *Longitude* (longitude where the airport is).

```
Code,Name,City,Country,Latitude,Longitude
CDG,Charles De Gaulle,Paris,France,49.012779,2.550000
ORY,Orly,Paris,France,48.725278,2.359444
LGA,La Guardia,New York,United States,40.777245,-73.872608
JFK,John F Kennedy Intl,New York,United States,40.639751,-73.778925
LHR,Heathrow,London,United Kingdom,51.477500,-0.461389
```

- File **airlines.csv**: contains the information of *444 airlines*.

  The first line includes the headers: *Code* (unique ICAO airline code), *Name* (official airline name), *Callsign* ("nickname" of airline, if there is no nickname, this field only contains "_"), *Country* (country of registry).

```
Code,Name,Callsign,Country
IBE,Iberia Airlines,IBERIA,Spain
KLM,KLM Royal Dutch Airlines,KLM,Netherlands
SWR,Swiss International Air Lines,SWISS,Switzerland
AAL,American Airlines,AMERICAN,United States
RYR,Ryanair,RYANAIR,Ireland
```

- File **flights.csv**: contains the information of *63832 flights*.

  The first line includes the headers: *Source* (IATA code of source airport), *Target* (IATA code of target airport), *Airline* (ICAO airline code).

```
Source,Target,Airline
CDG,JFK,AAL
JFK,CDG,AAL
ORY,LHR,IBE
ORY,LHR,BAW
OPO,LIS,TAP
```

  Note: connections are directed and there may be several flights with the same destination and origin, if they are from different airlines.

Statement of Work (SoW)

In order to develop your flight management system, you should meet the following important requirements, along with other functionalities you may find relevant.

1. **Read and parse the provided data**. This includes loading the airports, airlines and connections into an appropriate graph data structure. You may create one or more graphs that allow for the implementation of the required management system features. Note that for this project **it is mandatory to use the data structure provided for the practical classes to represent and CRUD graphs** (i.e., the *Graph class*).

2. **Develop a flight management system**. The system must incorporate a comprehensive set of functionalities that provide a detailed view of the global connectivity of the air travel network, and also ensure that users can make informed decisions when searching this network. Therefore, the system must include all the functionalities specified for this project and any others deemed relevant. It must also have a **user-friendly menu**, displaying the features you have implemented and their corresponding outputs in a clear, organized, and logical manner.

3. **Calculate and list statistics of the network**, including existing airports and connections. The following statistics are mandatory:
    i. Global number of airports and number of available flights;
    ii. Number of flights out of an airport; and from how many different airlines;
    iii. Number of flights per city/airline;
    iv. Number of different countries that a given airport/city flies to;
    v. Number of destinations (airports, cities or countries) available for a given airport;
    vi. Number of reachable destinations (airports, cities or countries) from a given airport in a maximum number of X stops (lay-overs);
    vii. Maximum trip and corresponding pair of source-destination airports (or pairs, if more than one), that is, the flight trip(s) with the greatest number of stops in between them;
    viii. Identify the top-*k* airport with the greatest air traffic capacity, that is, with the greatest number of flights;
    ix. Identify the airports that are essential to the network's circulation capability (airports are essential if, when removed, areas of the network start to be unreachable).

4. **Present the best flight option** (or the set of options if they are equivalent) for a given source and destination locations. The best flight option is considered to be the one with the least amount of stops. Locations may be specified by the user (inputs) using different criteria:
    i. Airport code or name;
    ii. City name, encapsulating all the airports departing from a given city;

iii. Geographical coordinates, considering the closest airport to the given coordinates. If there is more than one, consider all the closest ones.

The user may also wish to use combinations of these, that is, the system must allow the user to consider different criteria, for example, requesting a flight option *from an airport to a city*, or *from a city to a location*, or other possible combinations. If you wish to use distances in your program, make use of the Haversine Distance Function (Haversine Formula / implementation example).

5. **Search for the best flight option with filters** (that is, using only a subset of nodes/edges of the network) that respect user preferences. Some users may wish to travel using only one or a set of given airlines. Other users may want to minimize the number of different airlines in their travel. The system must present the best flight option (or the set of options if they are equivalent) taking into consideration the choices of the user.

6. Include **documentation** for the most relevant functions that you implemented, generated using Doxygen. Indicate the **time complexity** of the most relevant functions or algorithms in your program.

Expected Results

Your implementation should allow for storing and managing several different entities and their relationships, making use of an appropriate graph definition (based on the given Graph class) and the data structures that you consider relevant. In this sense, you should pay close attention to the following mandatory items:

- Create and maintain an appropriate **graph class** to achieve the desired operations;
- Allow constant-time **lookup** for airlines, airports and cities;
- Mind the **complexity** of your algorithms so that the program does not take too long to compute any of the required operations;
- Include **documentation** of the implemented code (generated using Doxygen) and indicate the **time complexity** of the most relevant functions or algorithms of your program.

Your program should also make use of important graph algorithms such as:

- Breadth and depth-first search;
- Connected components;
- Articulation points;
- …

Demonstration and Presentation

Preparing a concise and focused presentation is a very important skill. Therefore, you should structure a short 10 to 15-minute presentation of your work, focusing on the following points:

- Present your flight management system, using illustrative examples to demonstrate the implemented functionalities;
- Highlight your graph class, explaining the conceptual decisions that were made;
- Highlight and justify how you obtained constant time lookup for airports and cities;
- Highlight the most important aspects of your implementation.

You should also elaborate a **PowerPoint presentation** to support your presentation. **Instructions** for preparing the PowerPoint presentation are available on **Moodle**.

Turn-In Instructions and Deadline

Submit a zip file named **AED2324_PRJ2_G<TN>.zip** on Moodle, where **TN** stands for your class and group numbers (e.g., G**10** represents group **0** of class 2LEIC**01**), with the following content:

- Code folder, containing the program source code.
- Documentation folder, containing HTML documentation, generated using Doxygen.
- Presentation file (**PDF format**) that will serve as a basis for the presentation.

No submissions will be accepted 24 hours after the deadline. Exceptions apply for justified and documented technical submissions issues.

The deadline is the 2nd of January of 2024 at 8:30 AM.