

2019 Career Fair Programming Competition

Introduction

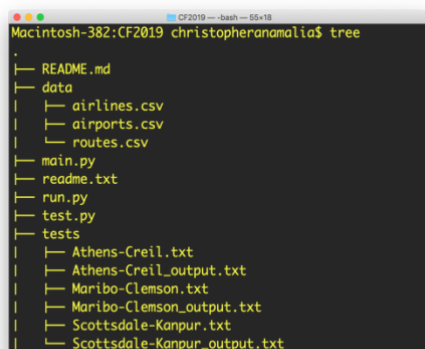
This challenge was related to the aviation sector. This program finds the optimal routes a user can use to get to a destination given the start city and destination city. The program also considers various airports and airlines in the country.

Approach to the problem

In approaching this problem, we considered the fact that given a start and destination city, there is sometimes a straight flight or a common point(s) of intersection if we approach from both sides. This program capitalized on that to get the routes to a destination. For selected routes, we use the Haversine formula to calculate the distance and find the flight with the shortest possible route if there are more than one flight. In the instance where the program identifies only one route to the place, the journey is automatically optimized by flight (Number of flights).

For the sake of optimality and faster runtime of code, the following measures are taken:

- We used python Object-Oriented-Programming
- Code was written in chunks as methods or functions
- Data size in csv file was shrunk to reflect the relevant ones. The means that only useful data was taken after reading. (Example the use of only active airline)
- Used inbuilt functions as much as possible since they usually written with best algorithm
- Used 'divide and conquer' method in looping though for-loops
- Deleted unnecessary variables in the function.

A terminal window on a Macintosh-382:CF2019 machine showing the file structure of the program. The command 'tree' has been executed, displaying a directory tree. The root directory contains a README.md file, a data folder, a main.py file, a readme.txt file, a run.py file, a test.py file, and a tests folder. The data folder contains airlines.csv, airports.csv, and routes.csv. The tests folder contains Athens-Creil.txt, Athens-Creil_output.txt, Maribo-Clemson.txt, Maribo-Clemson_output.txt, Scottsdale-Kanpur.txt, and Scottsdale-Kanpur_output.txt.

```
Macintosh-382:CF2019 christopheranamalia$ tree
.
├── README.md
├── data
│   ├── airlines.csv
│   ├── airports.csv
│   └── routes.csv
├── main.py
├── readme.txt
├── run.py
├── test.py
├── tests
│   ├── Athens-Creil.txt
│   ├── Athens-Creil_output.txt
│   ├── Maribo-Clemson.txt
│   ├── Maribo-Clemson_output.txt
│   ├── Scottsdale-Kanpur.txt
│   └── Scottsdale-Kanpur_output.txt
```

On the left indicates the file structure of the program. Most importantly, the program is `main.py`. All the csv files that run the program are in the data folder. The program code is in `main.py`. `test.py` is used for running tests on the module based on randomization. Its output and the corresponding input files are stored in the tests folder.

NOTE: In the output file, when optimality criteria is flights, it means the program selected the least and only number of flights to use among the lots. And if it is distance, it means it used the routes with the shortest distance.

References

Stack Overflow (2018). [http: https://stackoverflow.com/](https://stackoverflow.com/) - Gave us a little insight into debugging some of the errors we had.