DASC 5300/CSE 5300
Foundations of Computing
Instructor: Sharma Chakravarthy
Project I: Python Programming and Census Analysis

Made available on:	**10/4/2021**
Complete Project Due on	**10/28/2021 (by 11:59 pm)**
Submit to	**Canvas (uta.instructure.com)**
	1 zipped folder containing all the files/sub-folders
	Late submissions have a penalty as indicated!
Weight:	**15% of total**
Total Points:	**100**

This project continues the data analysis you started using a different data set and with different objectives. The goal is to use real-world data sets, similar to the ones you will encounter in your career. In this project, you will analyze an airlines data set to understand important things about airline operation. As usual, there will be pre-processing involved in this analysis as well.

By now, you should have Python installed on your machine and gotten your beak wet. You should also be comfortable with either Google colab or an IDE. For most of these class projects, Google colab is more than adequate as well as much easier and straightforward to use.

I. Problem Statement:

For this problem, you are given a large airlines data set indicating the flights operated by that airline. We have replaced the airline name with AL1, AL2, etc. It has about 6700 routes of several International and US carriers. The data set contains the following information in each line, separated by comma

Airline 2-letter (IATA) or 3-letter (ICAO) code of the airline.
Airline ID Unique OpenFlights identifier for the airline.
Source airport 3-letter (IATA) or 4-letter (ICAO) code of the source airport.
Source airport ID Unique OpenFlights identifier for source airport.
Destination airport 3-letter (IATA) or 4-letter (ICAO) code of the destination airport.
Destination airport ID Unique OpenFlights identifier for destination airport (see Airport)
Codeshare "Y" if this flight is a codeshare (that is, not operated by Airline, but another carrier), empty otherwise.
Stops Number of stops on this flight
Equipment 3-letter codes for plane type(s) generally used on this flight, separated by spaces

We have anonymized the airlines so you can try to identify the airline as part of your analysis.

II. What you need to do (revised)

1. Separate the airline that you are asked to analyze. There will be a spreadsheet indicating which airline you will analyze.

   a. Construct a graph (or a graph data structure) for the airline being analyzed by you. With airport code as the vertex, draw an undirected edge between the two airports in each line of the input. Since many Python packages/algorithms only accept integers as vertex, you may have to map the 3-letter airport code into an integer. You can use a dictionary or something else for this purpose.

      Important:

      i. The routes file contains routes for BOTH directions between airports. You need to keep only one of them as part of graph generation

      ii. Most packages that process graphs require node or vertex to be integers. So, you may have to map airport code into integers for processing

      iii. We are also providing a mapping from airport code to the city and country to be used AFTER figuring out the top-k hubs

   b. Using the given method, generate a few characteristics for this airline route graph. This will help understand some of the features of the data set you are analyzing. Interpret them from an airlines operation perspective.

   c. Find the top 3 to 5 nodes (or airports; top k, in general) from where there are more number of flights than other nodes for that airline. These are termed hubs by an airline. You can use a node centrality detection algorithm/package for this

   d. Once you have the top k hubs, try to identify the airline that you are working on. You can use any data that is available on the Internet for this purpose. We will give you a file that maps airport code to airport names and cities. You can search for airline hubs based on the cities you identify to figure out the airline.

   e. Once you have the airline identified, verify that with us. In this step of the analysis**,** your goal is to identify or predict the next hub for airline expansion. Typically, an airport already used by the airline is considered for the next hub. Since there are a lot of them, choosing the right one requires some additional analysis. For this, you can use demographic and other information (e.g., regional) that is available on the web.

      The purpose of this step is for you to develop a feel for the analysis beyond data given toyou. We will grade you based on the ideas you use/come up with for this analysis. The actual airport identified as the next hub is secondary. If it matches with the actual one, that is great. You need to argue why your process and the publicly available data you have used are appropriate.

You can visualize or plot the airline route map as a graph by using one of the python libraries.

III. Project Report

Please include the following sections in a **REPORT {.doc format}, in addition to the rest,** which you will turn in with your code:

- **Overall Status**
  Give a *brief* overview of how you went about approaching/solving the problem and doing this project. If you were unable to finish any portion of the project, please give details about what has been completed and your understanding of what is not completed. (This information is useful when determining partial credit.)

  If you had difficulty with Python, please clearly indicate them and what additional information you needed. I will use this as feedback to revise what is taught for the next offering.

- **File Descriptions**
  List any new files you have created or used and *briefly* explain their major functions and/or data structures. If you have added additional test cases, please summarize them using tables.

- **Division of Labor**
  Describe how you divided the work (for teams), i.e., which group member did what. Please also include how much time the team spent on this project. (This has no impact on your grade whatsoever; we will only use this as feedback in planning future projects -- so be honest!)

- **Problems encountered and how you handled them**
  List at least 3 problems you encountered (not syntax-related, preferably analysis- or logic-related) during the completion of the project. Choose those that challenged you. This will provide us some insights into how we can improve the description and forewarn students for future assignments.

- **The report should be an analysis report. You should reference publicly available data you have used for your analysis. Please limit your report to a maximum of 10 pages, including plots, graphs, explanation of analysis, and conclusions drawn. Anything beyond 10 pages will be ignored and not graded.**

Use milestones to pace yourself. Decomposition of the problem into sub-problems has been done for that purpose. Hence, this project lends itself to that. Each of the sub-problem can be a milestone. The last sub-problem (e) is likely to take some effort, so finish others early.

IV. What to Submit

- After you are satisfied that your project is complete and you are happy with the analysis, you upload it to canvas for grading. Please submit your **project report** and a table of routines/algorithms used/developed **in ONE zipped folder**. It may have sub-folders (one for each sub problem, for example.)
- All the above files should be placed in a single zipped folder named 'DASC5300_**Proj2_Fall21_team_<teamNo>**'. **Only one zipped folder should be uploaded using canvas.**
- You can submit your zip file at most 3 times. The latest one (based on timestamp) will be used for grading. So, be careful what you turn in and when!
- **Only one person per team should upload the zip file!**
- To discourage late submissions, a penalty of 20% per day (no partial penalty) will be applied. This means that if your submission is delayed by more than 5 days, do not bother submitting. We certainly do not want this delay to hurt your next project!

V. Coding Style:

If you write code in Python, please follow the coding guidelines for Python. We have discussed a number of do's and don'ts in the class. Please follow them. Python supports Pydoc (similar to JavaDoc) and I recommend you use it (for bonus 5 points) as well as comments in the code.

VI. Grading Scheme for the Completed Project:

The project will be graded using the following rubric. The report should contain a section of analysis for each item below and the team should be able to answer questions on how they arrived at this analysis:

Filtering your airline		5
Sub-problem a.	creating graph data structure	5
Sub-problem b.	Generating graph characteristics	10
Sub-problem c.	generating top k hubs	10
Sub-problem d.	Identify the airline	15
Sub-problem e.	Identifying/predicting the next hub	25
1. Q/A performance during demo		20
2. Challenges encountered and solution		10
	TOTAL (100)	**100**

There will be separate provisions on Canvas for submitting projects on time and with delay as Canvas closes the submission after the deadline. Please keep this in mind.

You are welcome to use your laptop (windows, apple, or Linux). It is your responsibility to have it working in your environment. You cannot debug code and fix problems during the demo! Any code you have written for the project should be included in the uploaded folder as a src folder. The timestamp of the submitted code and demo code should be identical. If not, penalty will be applied as stated above!