BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Work Integrated Learning Programmes Division
M.Tech (Data Science and Engineering)

(S1-21_DSECLZG519)
(Data Structures and Algorithms Design)
Academic Year 2020-2021

Assignment 1 – PS15 - [Food Delivery] - [Weightage 12%]

1. Problem Statement

Assume you work for a start-up company whose business model is to make available local delicacies across India. They work on the promise of same day delivery of food items across India and are trying to build a logistical chain for the same which can deliver across Airports using their patented food boxes which preserve the food quality within transit. Your team is tasked with developing a system which can record all air cargo delivery routes that are available between airports in India. The data is captured such that each cargo plane and its associated airports are captured as vertices and the association as edges. Assume that the flights are bi-directional, that means the same cargo flight number is used for the onward and return journey.

As a first phase, they want to identify the following information.

1. List of unique cargo planes and list of unique airports that have delivery service.
2. Find out which airport is the main transport hub. (Airports which is visited by the greatest number of cargo planes)
3. Find out which airports are connected by a single cargo flight.
4. If a food box needs to be sent directly from airport a to airport b, which flight should they book?
5. Can a food box be sent from airport a to airport b even if it must be transferred (change flight) at an intermediary airport c (any number of transfers).

Requirements:

1. Implement the above problem statement in Python 3.7 using graph data structures and adjacency matrix. Do not use the inbuilt Python data structure.
2. Perform an analysis for the features above and give the running time in terms of input size: n.

Operations:

1. **def readAirportFlightfile(self, inputfile)**: This function reads the input file **inputPS15.txt** containing the name of the airport and the cargo flights between them in one line separated by a slash. A sample input file entry is shown below. The flight number is the first entry in each row followed by the different airports it services separated by a slash '/'

   indigo777 / Chennai / New Delhi

   The function should create relevant vertices for both the cargo flights and its associated airports and relevant edges to indicate the association of a flight and its connecting airports. Ensure that the vertices are unique and there are no duplicates.

2. **def showAll(self):** This function displays the count of unique cargo flights and airports entered through the input file. It should also list out the unique cargo flights and airports that have cargo service stored. This function is called after all input data has been captured. The output of this function should be pushed into **outputPS15.txt** file. The output format should be as mentioned below.

   --------Function showAll --------
   Total no. of cargo flights: 5
   Total no. of airports: 8
   List of Cargo flights:
   Indigo777
   Spicejet444
   GoAir111
   Indigo333
   AirIndia999

   List of airports:
   Chennai
   New Delhi
   Ahmedabad
   Mumbai
   Nagpur
   Hyderabad
   Calcutta
   Vishakhapatnam

   ---

3. **def displayHubAirport(self):** This function displays the name of the airport which is visited by the most number of cargo flights. The function also displays the names of the incoming cargo flights to the **outputPS15** file. The function is triggered when the 'searchHubAirport' tag is found in the file **promptsPS15**.txt file.

   searchHubAirport:

   The output of this function should be appended into **outputPS15.txt** file. The output format should be as mentioned below.

   --------Function displayHubAirport --------
   Main hub airport: New Delhi
   Number of cargo flights visited: 3
   List of Cargo Flights:
   Indigo666
   AirIndia111
   GoAir222

   ----

4. **def displayConnectedAirports(self, flight):** This function displays all the airports are connected by a single cargo flight. The function reads the input cargo flight number from the file **promptsPS15.txt** with the tag as shown below.

   searchFlight: Indigo666

   searchFlight: AirIndia111

   The output of this function should be appended into **outputPS15.txt** file. If a flight is not found, an appropriate message should be output to file. The output format should be as mentioned below.

   --------Function displayConnectedAirports --------
   Cargo flight number: Indigo666
   Number of airports connected: 3
   List of airports connected directly by : Indigo666
   Ahmedabad
   Mumbai
   Nagpur

   ----

5. **def displayDirectFlight(self, airport a, airport b):** This function displays the cargo flight name which can be booked to send a food box directly from airport a to airport b. The function reads the input airports from the file **promptsPS15.txt** with the tag as shown below.

searchAirports: Calcutta: New Delhi
searchAirports: Chennai: Hyderabad

The output of this function should be appended into **outputPS15.txt** file. If there is no direct flight or an airport is not found, an appropriate message should be output to the file. The output format should be as mentioned below. If there is more than one flight that can be booked, the flight number you encounter first can be output.

--------Function displayDirectFlight --------
Airport A: Calcutta
Airport B: New Delhi
Food box can be sent directly: Yes, Indigo666 (if no, display appropriate message)

---

6. **def findServiceAvailable(self, airport a, airport b):** This function finds whether a food box can be sent from airport a to airport b with any number of stops/transfers (i.e. to deliver the food box from airport a to airport b it might even get transferred on another flight at an intermediary airport c). The function reads the input airports from the file **promptsPS15.txt** with the tag as shown below.
   ServiceAvailability: Calcutta: Mumbai
   ServiceAvailability: Nagpur: Vishakhapatnam

   Also display the entire route to transfer the food box from airport a to airport b. The output of this function should be appended into **outputPS15.txt** file. If the food box can't be transferred or an airport is not found, an appropriate message should be output to the file. The output format should be as mentioned below.

   --------Function findServiceAvailable --------
   Airport A: Calcutta
   Airport B: Nagpur
   Can the food box be sent: Yes, Calcutta > Indigo666 > New Delhi > AirIndia111 > Ahmedabad > GoAir444 > Nagpur (if no, display appropriate message)

   ---

7. Include all other functions that are required to support these basic mandatory functions.

Sample file formats

Sample Input file

The input file **inputPS15.txt** contains names of the flights and the connected airports in one line separated by a slash (/).

Sample inputPS15.txt

```
Indigo666 / Chennai / New Delhi
Indigo777 / Calcutta / New Delhi
Spicejet222 / Ahmedabad / Nagpur / Mumbai
AirIndia111 / Ahmedabad / New Delhi
Vistara555 / Vishakhapatnam / Hyderabad
```

Sample promptsPS15.txt

```
searchHubAirport
searchFlight: Indigo666
searchFlight: AirIndia111
searchAirports: Calcutta: New Delhi
searchAirports: Chennai: Hyderabad
ServiceAvailability: Calcutta: Mumbai
ServiceAvailability: Nagpur: Vishakhapatnam
```

Sample outputPS15.txt

```
--------Function showAll --------
Total no. of cargo flights: 5
Total no. of Airports: 8

List of Cargo flights:
AirIndia111
Indigo666
.
.
List of Airports:
Chennai
New Delhi
Ahmedabad
.
.
---------------------------------------

--------Function displayHubAirport --------
Main transport hub: New Delhi
Number of flights visited: 3
List of Cargo flights:
AirIndia111
Indigo666
Indigo777
---------------------------------------
```

--------Function displayConnectedAirports --------
Cargo flight name: Indigo666
Number of Airports connected: 2
List of Airports connected directly by Indigo666:
Chennai
New Delhi

--------Function displayDirectFlight --------
Airport A: Calcutta
Airport B: New Delhi
Food box can be sent directly: Yes, Indigo777 (if no, display appropriate message)

… other function outputs...

Note that the input/output data shown here is only for understanding and testing, the actual file used for evaluation will be different.

2. Deliverables

1. Word document **designPS15_<group id>.docx** detailing your design and time complexity of the algorithm.
2. **[Group id]_Contribution.xlsx** mentioning the contribution of each student in terms of percentage of work done. Download the Contribution.xlsx template from the link shared in the Assignment Announcement.
3. **inputPS15.txt** file used for testing
4. **prompts15.txt** file used for testing
5. **outputPS15.txt** file generated while testing
6. **.py file** containing the python code. Create a single *.py file for code. Do not fragment your code into multiple files

Zip all of the above files including the design document and contribution file in a folder with the name:

[Group id]_A1_PS15_FoodDelivery.zip and submit the zipped file.

Group Id should be given as **Gxxx** where xxx is your group number. For example, if your group is 26, then you will enter G026 as your group id.

3. Instructions

1. It is compulsory to make use of the data structure(s) / algorithms mentioned in the problem statement.
2. Ensure that all data structure insert and delete operations throw appropriate messages when their capacity is empty or full. Also ensure basic error handling is implemented.

3. For the purposes of testing, you may implement some functions to print the data structures or other test data. But all such functions must be commented before submission.
4. Make sure that your read, understand, and follow all the instructions
5. Ensure that the input, prompt and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.
6. The input, prompt and output samples shown here are only a representation of the syntax to be used. Actual files used to evaluate the submissions will be different. Hence, do not hard code any values into the code.
7. Run time analysis is to be provided in asymptotic notations and not timestamp based runtimes in sec or milliseconds.
8. Please note that the design document must include
    a. The data structure model you chose with justifications
    b. Details of each operations with the time complexity and reasons why the chosen operations are efficient for the given representation
    c. One alternate way of modelling the problem with the cost implications.
9. Writing good technical report and well document code is an art. Your report cannot exceed 4 pages. Your code must be modular and quite well documented.

Instructions for use of Python:

1. Implement the above problem statement using Python 3.7.
2. Use only native data types like lists and tuples in Python, do not use dictionaries provided in Python. Use of external libraries like graph, numpy, pandas library etc. is not allowed. The purpose of the assignment is for you to learn how these data structures are constructed and how they work internally.
3. Create a single *.py file for code. Do not fragment your code into multiple files.
4. Do not submit a Jupyter Notebook (no *.ipynb). These submissions will not be evaluated.
5. Read the input file and create the output file in the root folder itself along with your .py file. Do not create separate folders for input and output files.

4. Deadline

1. The strict deadline for submission of the assignment is **Wednesday, 22nd Dec, 2021.**
2. The deadline has been set considering extra days from the regular duration in order to accommodate any challenges you might face. No further extensions will be entertained.
3. Late submissions will not be evaluated.

5. How to submit

1. This is a group assignment.
2. Each group has to make one submission (only one, no resubmission) of solutions.
3. Each group should zip all the deliverables in one zip file and name the zipped file as mentioned above.
4. Assignments should be submitted via Canvas > Assignment section. Assignment submitted via other means like email etc. will not be graded.

6. Evaluation

1. The assignment carries 12 Marks.
2. Grading will depend on
    a. Fully executable code with all functionality working as expected
    b. Well-structured and commented code
    c. Accuracy of the run time analysis and design document.
3. Every bug in the functionality will have negative marking.
4. Marks will be deducted if your program fails to read the input file used for evaluation due to change / deviation from the required syntax.
5. Use of only native data types and avoiding libraries like numpy, graph and pandas will get additional marks.
6. **Plagiarism will not be tolerated. Copy / Paste's from web resources / or your friends' submission will attract severe penalty to the extent of awarding 0 marks. We will not measure the extent of such blatant copy pastes and details of who copied from whom and such details while awarding the penalties. It's the responsibility of the team to solve and protect your original work.**
7. Source code files which contain compilation errors will get at most 25% of the value of that question.

7. Readings

Text book: Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition). **Chapters:** 6