CSE 551
FOUNDATION OF ALGORITHMS
Homework 4- Programming Assignment
Fall 2019

Instructor : Dr.Anurbha Sen

Capacity of NAS

QUESTION :

Given: - Source city of the National Airspace System is Los Angeles (LAX) and the Destination city is New York city (JFK) .Static data need to computed regarding the operation of three airlines American Airlines (AA), Delta Airlines (DL) and United Airlines (UA) on 6th January 2020 with starting time 12:00 AM and ending at 11:59 PM(Total duration is considered only for one day). Intermediate stops that can be considered as par of the problem are San Francisco (SFO), Phoenix (PHX), Seattle (SEA), Denver (DEN), Atlanta (ATL), Chicago (ORD),Boston (BOS) and Washington DC (IAD). .

To Find: - Compute the capacity of the NAS from LAX to JFK on 6th January 2020(12.00 AM to 11.59PM).

Solution: - Capacity Computation Tool Design

Inputs to the Algorithm :

File : FOA_HW_DATASET4.csv

Source	Destination	Start time	End time	Capacity	Flight Number
LAX	JFK	2:15 PM	10:31 PM	Boeing 767-300	DL 2262
LAX	JFK	12:40 PM	9:02 PM	Boeing 767-300	DL 468
LAX	JFK	11:25 AM	7:54 PM	Boeing 757-200	DL 2787
LAX	JFK	2:00 PM	10:18 PM	Airbus A321	AA 32
LAX	JFK	12:00 PM	8:24 PM	Airbus A321	AA 274
LAX	JFK	6:05 AM	2:23 PM	Boeing 767-300	DL 1436
LAX	JFK	9:25 AM	5:50 PM	Boeing 767-300	DL 2164
LAX	JFK	11:00 AM	7:25 PM	Airbus A321	AA 4
LAX	JFK	7:55 AM	4:12 PM	Boeing 767-300	DL 1258
LAX	JFK	7:00 AM	3:18 PM	Airbus A321	AA 118
LAX	JFK	10:00 AM	6:23 PM	Airbus A321	AA 238
LAX	JFK	8:00 AM	4:19 PM	Airbus A321	AA 2
LAX	JFK	12:30 AM	8:48 AM	Airbus A321	AA 292
LAX	SFO	8:54 AM	10:36 AM	Boeing 737-800	UA 488

The input to the capacity computation tool is as follows:

- **Source** : Source is the city where the flight depart.

- **Destination :** Airport/City name where the flight arrives.

- **Start time**: Time when the flight departure from the terminal.

- **End time** : Time when the flight land at the terminal.

NOTE : Both Start and End times can be in either 12 hour format : 2:16AM or 1.45 PM or can be in 24 format :13:00. If entered in wrong format will raise a Value Error telling the user about the issue.

- **Capacity:** Capacity that the flight can accommodate. Can be in two formats : 1) 300 2) Boeing 737-800. Tool will generate the seating capacity from the input given using a static lookup table and compute the capacity.

- **FlightNumber:** Aircraft Number . Can be from AA,UA or DL.

Platform : Python 3.7 - Anaconda(Spyder)

Definitions :

- **Construction of Graph**: - The problem consists of handling both the spatial(city names) and temporal information(Time landed/departure) in a single node. A graph is constructed in such a way that both these spatial and temporal information is captured. As considering only either spatial or temporal information can give wrong results. A dictionary data structure is used where the key ranges from 0 to 23, which represents 24-hour time and each key consists of value as dictionary, which comprises of start and departure details, flight numbers and location, capacity. The time is rounded off to the nearest hour(1.00 - 1.29 is rounded off to 1and 1.30 to. 1:59 is updated to 2). All these data is retrieved from the the csv file provided in the zip file.

- **Ford – Fulkerson Algorithm**: -It is the algorithm that computes maximum flow in the network. The idea behind the algorithm is find the capacity as long as flow exists between the source and sink. I used the same logic behind this algorithm for solving this problem but all constraints wrt Ford-Fulkerson is not applicable in this algorithm.

- **Augmented Path**: - A path with no cycles and edges with positive capacity from source to sink. In other words, path can be enhanced or augmented.

- **Residual Graph**: - A residual graph is a graph with possible additional flow.

Pseudo Code :

- **Algorithm**: - Once the graph is obtained. The ford Fulkerson algorithm can be applied to obtain maximum flow in the network by using the concepts of augmented path and residual graph.

1. Initialize the source city to “LAX” and Iterate through all the time nodes from 0 to 23, at every hour, and look for the time where there is a flight from LAX to some other location.

2. Once the particular path is identified iterate further looking for the time node where the flight started from LAX landed. Once the time node is identified, update the particular capacity with respect to that edge. Then update the source and destination cities at that particulate node and look over for the flights from that particular source.

3. If the path from intermediate node to sink is broken then that particular path is eliminated from the list to avoid revisiting and save some time.

4. A visited list is maintained in order to avoid repeated paths.

5. Finally the total capacity is computed by adding the capacities over all the augment paths computed from LAX to JFK.

- **Methods**: -

time12to24hr – Utility that converts 12 hour to 24 hour time format and also round off time values

FlightMaxCapacity - Class which handles entire logic.

Capacity - Method in FlightMaxCapacity where the entire logic is imbibed.

- **Errors**: - Value Error is raised incase the input file path provided is in incorrect format.

- **Time Complexity :**

The key idea behind the algorithm developed for the tool is Ford Fulkerson where we iterate through all the nodes(24 nodes here) in the graph and generate the augmented paths. So we run a loop for every augmented path. The worst case complexity for Ford Fulkerson algorithm is O(maxFlow * Edges). In this case, the maxFlow computed is 6154 and the number of edges would be equal to number of total flights. Data set I generated comprises of 625 flights which would be the Edges value. Thus the complexity for this problem would be

O(maxFlow * Edges)

Where maxFlow = 6154 and Edges = 625

```
In [21]: import pandas as pd

In [22]: f = pd.read_csv(r"/Users/teja/Desktop/foa/Hw4_FOA/Final/foa.csv")

In [23]: f.shape
Out[23]: (625, 6)

In [24]:
```

Final Result :

The maximum passengers or capacity travelling from Los Angeles (LAX) to New York city (JFK) on 6th January 2020 between 12:00 AM to 11:59 PM is **6154**

```
In [24]: runfile('/Users/teja/Desktop/foa/FOAHW4.py', wdir='/Users/teja/
Desktop/foa')
Total Capacity:  6154

In [25]:
```