**Programming Assignment**

**CSE 551**

Ujwal Krishna Yerrapothu

1220305697

**LANGUAGE:** Python

**IDE:** Google collab

Files: flights.txt, NAS\_Max\_Flow.ipynb

Execution: Click the "upload files" link and choose Max Flow.ipynb and flight.txt.

Set of data The following details regarding a flight between two cities are contained in the text file flights.txt.

* Source
* Destination
* Departure
* Arrival
* Flight capacity

Decoding the code:

**getData():**

A list of Flight objects is created by reading flight-related data from a text file and using the getData() function. The airports from which a flight departs and arrives, the times at which it departs and arrives, and the number of passengers on board are all included in comprehensive detail in each Flight object.

**getPort():**

The time of day is represented by the integer a in the getPort() function, and the airport code is represented by the string d in the same function. An integer index in the flow network that corresponds to the airport and daytime are returned by the function.

To determine the index, the function makes use of a series of if statements. If the airport is LAX, the function returns 0. Whether the airport is SFO, PHX, SEA, DEN, ATL, ORD, BOS, or IAD, the function generates an index based on the time of day. These airports have an index that is a multiple of 24 times the time of day, so different times of the day are represented by different indices. For instance, the SFO index is equal to 10 at 9 a.m., the PHX index is equal to 25 at 9 a.m., and so on. If the airport is JFK, the function returns 193, which is the JFK index in the flow network. The capability returns - 1 to show that the given air terminal is invalid on the off chance that the given air terminal code doesn't relate to one of the perceived air terminals.

The breadth-first search (BFS) method for finding a path in a graph is put into action by the function BreadthFirstSearch(). The rGraph parameter is used to represent the graph, which is a two-dimensional array of integers. The parent boundary is an exhibit that will be utilized to hold the way that the BFS calculation finds, and the s and t boundaries sub for the source and objective hubs, individually.

The function returns True if the graph shows a path from s to t; If not, False is returned. The path can be recreated by using the parent array. The BFS algorithm first looks into all of the neighbors of the source node before moving on to the next level of neighbors. Until the algorithm either finds the target node t or determines that there is no path through the entire graph, this procedure is repeated. The path that is being examined is stored in the parent array as it is updated along the way.

The fordFulkersonMaxFlow() function implements the Ford-Fulkerson algorithm for calculating a flow network's maximum flow. The graph parameter is the two-dimensional array of integers that makes up the flow network. The parameters s and t represent the source and destination nodes, respectively. The Ford-Fulkerson algorithm finds an augmenting path from s to t repeatedly in the residual graph and increases flow along that path as it does so, returning the maximum amount of flow that can be delivered in the specified flow network. The residual graph is the result of modifying the original flow network to create it. In this graph, the capacities of the edges are updated as traffic increases along them.

The bfs() function is used by the function to locate an augmenting path in the residual graph. After determining the maximum flow that can be transferred along that path, the residual graph is updated. This process should be repeated until the residual graph no longer contains any enhancing pathways. The Adjacency Matrix:

The adjMat array, a two-dimensional number array, is used to represent the flow network. It is initially initialized as a square matrix, with all of its capabilities set to zero. This shows that there are no flights between the two airports at first.

The application then alters the capacities of the flow network's edges in accordance with the flight data by reading the data from the input text file using the getData() function. Using the getPort() method, the application determines the indices of the flow network's source and destination nodes for each flight. If LAX and JFK are the respective source and destination nodes, the program increases the capacity of the directFlow variable. If this is not the case, the program adds the capacity of the flight to that of the edge in the flow network between the source and destination nodes.

After all of the flight data have been analyzed, the adjMat array displays the entire flow network along with the updated capacity of its edges.

Main():

The function is a computer program that ascertains the maximum flow in a flow network using the Ford-Fulkerson approach. The maximum flow is the amount of passengers who can be transported from LAX (Los Angeles) to NY, and the flow network reflects flights between various airports (JFK) (New York). The adjMat array, which serves as the program's representation of the flow network, is initially started by setting all of its capacities to 0. The capabilities of the edges of the flow network are updated in light of the flight data after reading the data from the input text file with the getData() function.The flow network is then altered by setting some edges' capacities to the highest integer number that can be achieved. This is done to show that passengers can switch planes at any airport in the same city, but not at airports in other cities.

The fordFulkersonMaxFlow() method is then used by the application to determine the maximum flow in the updated flow network.

RESULT:

Based on the calculations above, the maximum number of passengers who can board a flight from Los Angeles to New York is 6258.