Explanation of the Algorithm: Minimum Trips Problem

Ghazanfar Ghaffar 22i-2079

Problem Context

The Minimum Trips Problem involves finding the fewest number of trips required to transport a given number of tourists from a starting city to a destination city using roads with defined capacities. This problem can be understood as a variation of the "maximum flow" problem in network theory, where each road has a certain capacity and the goal is to maximize the flow from the source to the sink (destination).

Solution Approach

The approach to solving the Minimum Trips Problem utilizes the concept of maximum flow in a flow network. Here's how the algorithm was structured and implemented:

Graph Representation:

Cities and roads between them are represented as nodes and edges in a graph, respectively.

Each edge has an associated capacity, which represents the maximum number of tourists that can travel that road in a single trip.

Maximum Flow Algorithm (Edmonds-Karp Implementation):

The core of the solution relies on finding the maximum flow from the start city to the destination city using the Edmonds-Karp algorithm, which is an implementation of the Ford-Fulkerson method using breadth-first search (BFS) to find augmenting paths.

The algorithm iteratively finds paths from the source to the sink where additional flow can be pushed and continues until no such paths exist.

For each path, the flow is increased up to the minimum capacity edge in the path, ensuring the flow never exceeds the given capacities of the roads.

Calculation of Minimum Trips:

Once the maximum flow is determined, the minimum number of trips required is calculated by dividing the total number of tourists by the maximum flow value (which represents the maximum number of tourists that can be transported per trip).

Since the number of tourists and maximum flow are integers, and trips can't be fractioned, the division rounds up, ensuring all tourists are transported.

Detailed Steps in the Code

Graph Construction:
A graph is constructed with nodes representing cities and edges representing roads with capacities.
Input is read from a file specifying the number of cities, number of roads, and for each road, the connected cities and its capacity.

Edge Addition with Reverse Capacity:
For each road, an edge is added in the graph along with a reverse edge initialized with zero capacity, necessary for the Edmonds-Karp algorithm to handle backflows effectively.

Maximum Flow Calculation:
Using BFS, augmenting paths from the source (start city) to the sink (destination city) are found iteratively.
Flow is adjusted along the path and reverse paths are updated to allow future adjustments in flow if necessary.

Edge Case Handling:
The solution checks if the calculated maximum capacity is zero (indicative of no valid path or capacity issues), which would make it impossible to transport any tourists. In such cases, the program outputs an appropriate message and terminates.

Conclusion

The algorithm efficiently solves the Minimum Trips Problem by leveraging network flow theory, specifically the Edmonds-Karp algorithm for maximum flow. The solution not only finds the maximum number of tourists that can be transported per trip but also calculates the minimum number of trips required to transport all tourists from a given start to a destination city. This approach is robust, handles edge cases, and provides a foundation that can be extended for more complex network flow problems.