

This code implements a solution for the Traveling Salesman Problem using a dynamic programming approach. The algorithm should find the shortest possible route that visits each city exactly once and returns to the origin city. The resulting tour should have the minimum total distance among all possible tours.

**How it works:**

1. The algorithm starts with a graph representing cities (vertices) and the distances between them (weighted edges).
2. It uses a recursive approach with memoization to explore all possible paths through the cities.
3. For each path, it calculates the total distance traveled.
4. The algorithm aims to find the path with the minimum total distance that visits each city exactly once and returns to the starting city.
5. The resulting path is returned as the solution to the Traveling Salesman Problem.

Currently, the implementation of this feature contains a logic bug that causes the algorithm to deviate from its intended functionality.

**One example of the algorithm's expected outcomes:**

- Path: New York -> Chicago -> Houston -> Phoenix -> Los Angeles -> New York
- Total Tour Distance = 5200

**One example of the algorithm's actual outcomes:**

- Path: New York -> Los Angeles -> Phoenix -> Chicago -> Houston -> New York
- Total Tour Distance: -5100