**EXPERIMENT 14 - Traveling Salesman Problem**

**Aim:**

Write a program to implement traveling salesman problem.

**Theoretical Background:**

The Traveling Salesman Problem (TSP) is a classic optimization problem in computer science and mathematics. Given a set of cities and the distances between each pair of cities, the goal is to find the shortest possible route that visits each city exactly once and returns to the starting city. The problem is considered NP-hard, meaning that finding the exact solution becomes increasingly difficult as the number of cities increases.

There are several approaches to solving the TSP, including exact algorithms, approximation algorithms, and heuristics. Exact algorithms aim to find the exact solution to the problem by systematically searching through all possible routes. These algorithms, however, can be computationally expensive and become infeasible for large problems. Approximation algorithms aim to find an approximate solution that is close to the optimal solution, while heuristics aim to quickly find a solution that is likely to be good, but may not be the optimal solution.

Some of the most commonly used approaches for solving the TSP include:

* Branch and Bound: This is an exact algorithm that prunes away large portions of the search space by eliminating routes that cannot be better than the best solution found so far.
* Genetic Algorithms: This is a heuristic approach that uses principles from evolution, such as natural selection and genetic variation, to search for a good solution.
* Greedy Algorithms: This is a heuristic approach that starts with an arbitrary city and repeatedly selects the nearest unvisited city as the next city to visit.
* Linear Programming: This is an optimization approach that can be used to find a solution to the TSP by formulating the problem as a linear program and using linear programming algorithms to solve it.

No single method is guaranteed to work best for all instances of the TSP, so finding the best solution often involves using a combination of methods or adapting the method to the specific problem at hand.

**Algorithm:**

Step 1: Start.

Step 2: Define the problem. Start by defining the cities, the distances between each pair of cities and the starting city. This can be represented as a graph, where the cities are nodes and distances are edges.

Step 3: Create an initial solution. This can be done by randomly generating a solution, or using an existing solution as a starting city and adding the nearest unvisited city to the route until all cities have been visited.

Step 4: Calculate the total distance of the route found in Step 3 to evaluate the quality of the solution.

Step 5: Improve the solution, this can be done by swapping cities in the route and evaluating whether the cost decreases. If it does, then the new solution is kept, otherwise the old solution is kept.

Step 6: Repeat steps 4 and 5 until a satisfactory solution is found or a stopping criterion is met.

Step 7: Return the best solution or the route with the shortest total distance found in step 6 as the solution to the TSP.

Step 8: Stop.

**Conclusion:**

Python program to implement traveling salesman problem is implemented successfully.