**Artificial Intelligence (MCA573D)**

**LAB-5**

**1. Algorithm –** Travelling salesman problem with backtracking and dynamic programming.

**2. Definition and Example -**

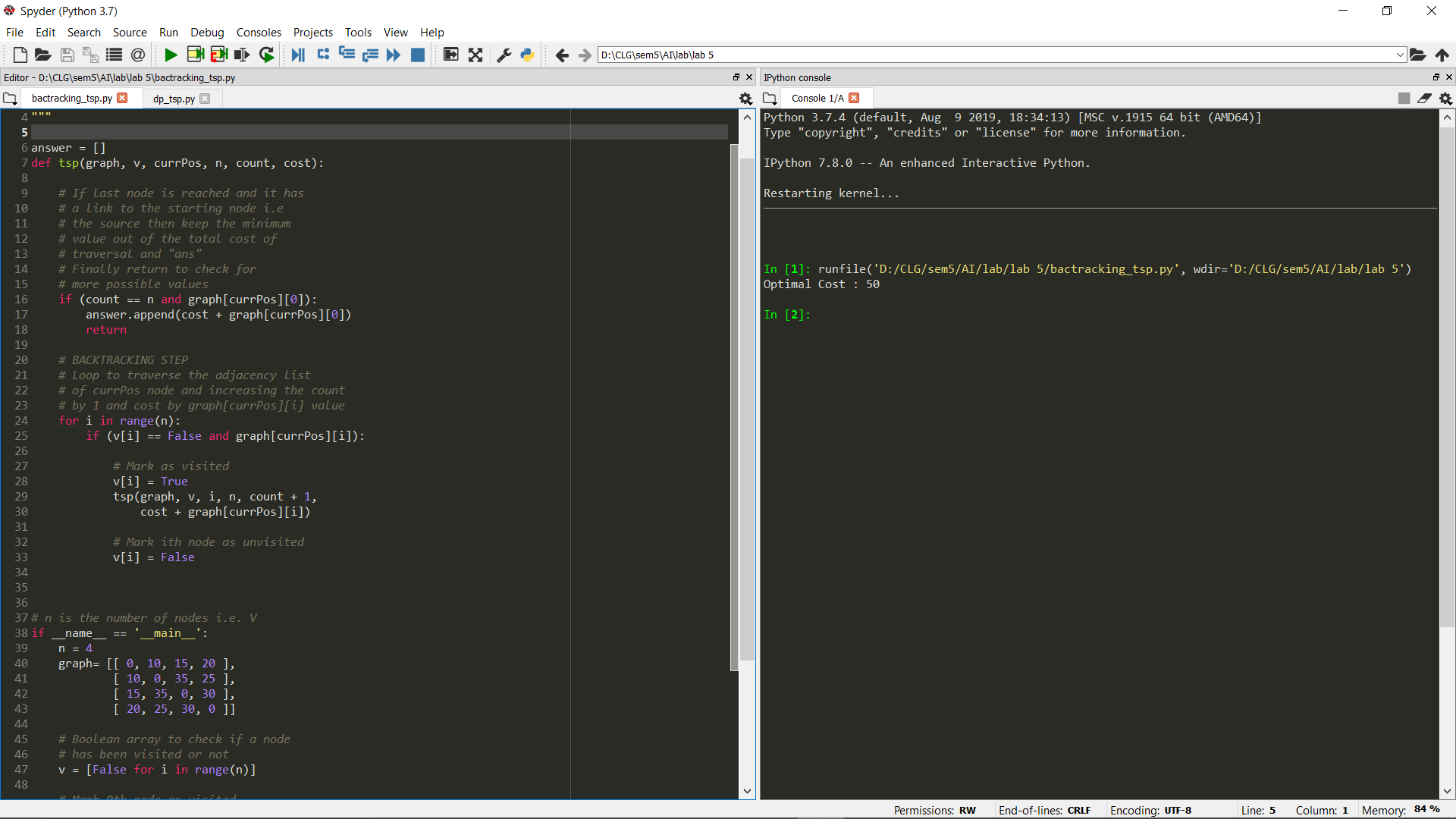
The traveling salesman problem (TSP) is an [algorithmic](https://whatis.techtarget.com/definition/algorithm) problem tasked with finding the shortest route between a set of points and locations that must be visited. In the problem statement, the points are the cities a salesperson might visit. The salesman‘s goal is to keep both the travel costs and the distance travelled as low as possible.

Focused on optimization, TSP is often used in computer science to find the most efficient route for data to travel between various [nodes](https://techtarget.com/searchnetworking/definition/node). Applications include identifying network or hardware optimization methods.  It was first described by Irish mathematician W.R. Hamilton and British mathematician Thomas Kirkman in the 1800s through the creation of a game that was solvable by finding a Hamilton cycle, which is a non-overlapping path between all nodes.

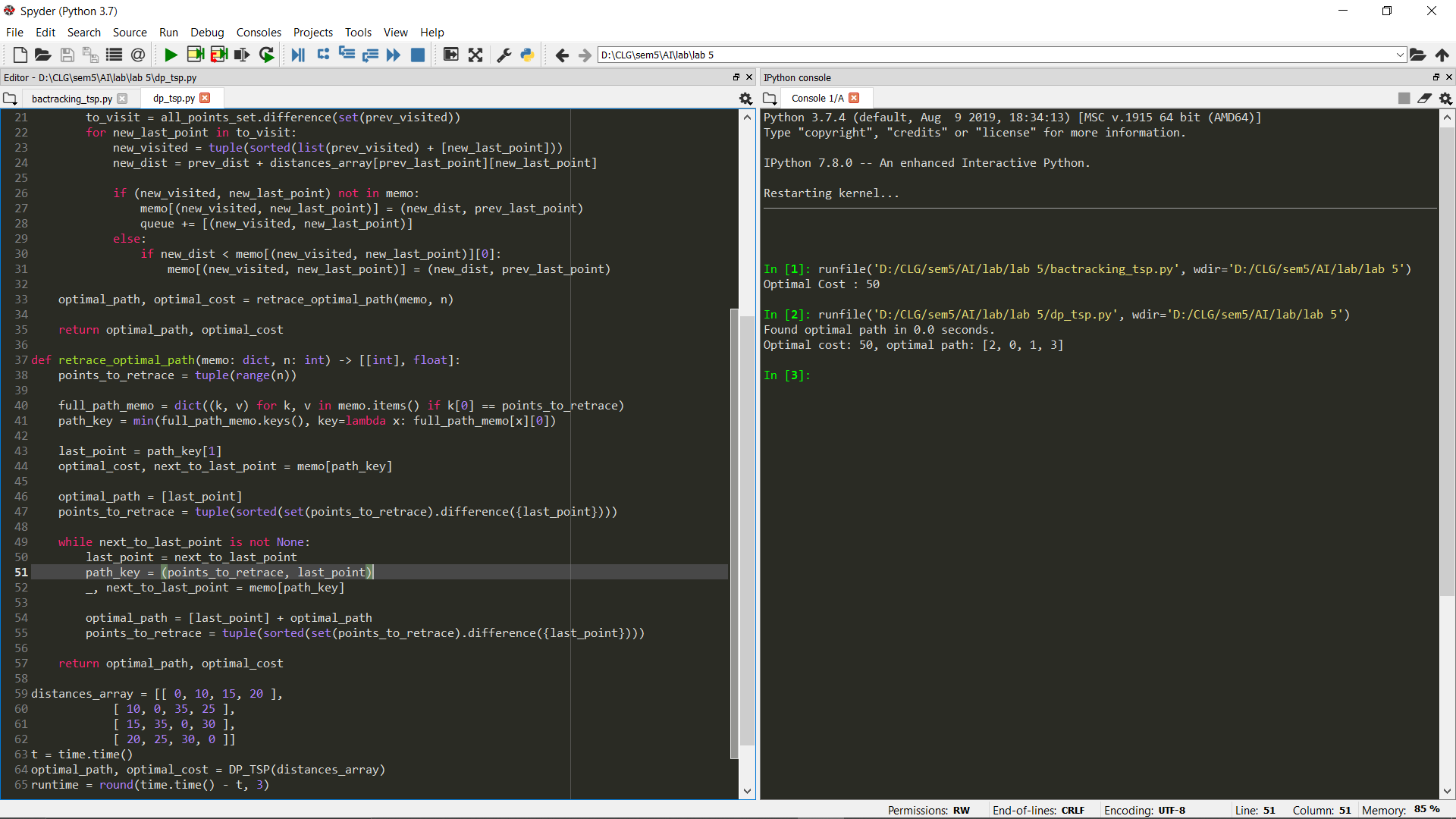
TSP has been studied for decades and several solutions have been theorized. The simplest solution is to try all possibilities, but this is also the most time consuming and expensive method.

**3. Implementation -**

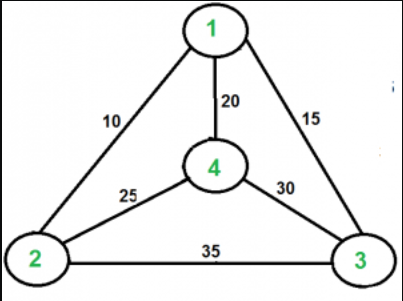
Following is the code implementation of TSP by backtracking in Python :



Following is the code implementation of TSP by dynamic programming in Python :



**4. Explanation –**

****

Following is the explanation of the python code used in implementation:

1. In backtracking we use a simple approach is which we use :

* Consider city 1 (let say 0th node) as the starting and ending point. Since route is cyclic, we can consider any point as starting point.
* Start traversing from the source to its adjacent nodes in dfs manner.
* Calculate cost of every traversal and keep track of minimum cost and keep on updating the value of minimum cost stored value.
* Return the permutation with minimum cost.

1. Where as in dynamic programming we use the following approach:

Initially, all cities are unvisited, and the visit starts from the city Let N. We assume that the initial travelling cost is equal to 0. Next, the TSP distance value is calculated based on a recursive function. If the number of cities in the subset is two, then the recursive function returns their distance as a base case.

On the other hand, if the number of cities is greater than 2, then we’ll calculate the distance from the current city to the nearest city, and the minimum distance among the remaining cities is calculated recursively.

Finally, the algorithm returns the minimum distance as a TSP solution.

Here we use a dynamic approach to calculate the cost function. Using recursive calls, we calculate the cost function for each subset of the original problem.

**5. Time Complexity –**

1. O(N!), As for the first node there are N possibilities and for the second node there are n–1 possibilities. For N nodes time complexity

= N \* (N – 1) \* . . . 1 = O(N!)

1. In the dynamic algorithm for TSP, the number of possible subsets can be at most N\*2^N. Each subset can be solved in O(N) times. Therefore, the time complexity of this algorithm would be O(N^2\*2^N).