LAB 3

In this lab, we implement two algorithms to solve travelling salesman problem:

1. Traveling Salesman Problem using Naive Algorithm.

1. Traveling Salesman Problem using Dynamic Programming Algorithm.

One graph was analyzed.

Given input and get output from both the Algorithm:

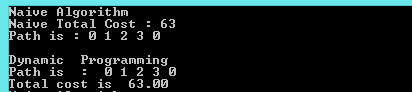
1. 4 nodes input

0.00,8.25,12.17,26.17

8.25,0.00,15.23,30.08

12.17,15.23,0.00,14.87

26.17,30.08,14.87,0.00

Output : 

1. 5 Nodes input

0.00,37.85,42.01,60.17,74.20

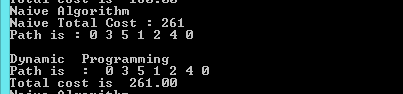
37.85,0.00,49.52,54.23,51.62

42.01,49.52,0.00,21.93,47.17

60.17,54.23,21.93,0.00,29.02

74.20,51.62,47.17,29.02,0.00

Output:



1. 6 nodes input

0.00,85.00,46.00,36.00,62.00,32.00

85.00,0.00,58.00,94.00,84.00,22.00

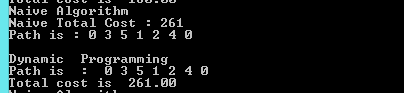
46.00,58.00,0.00,77.00,63.00,83.00

36.00,94.00,77.00,0.00,80.00,20.00

62.00,84.00,63.00,80.00,0.00,54.00

32.00,22.00,83.00,20.00,54.00,0.00

Output:



1. 8 nodes input

0.00,66.01,47.76,45.65,73.88,55.97,84.40,30.46

66.01,0.00,27.02,34.71,61.03,23.02,24.84,61.29

47.76,27.02,0.00,7.81,41.05,8.25,51.62,35.11

45.65,34.71,7.81,0.00,35.69,13.60,59.40,28.64

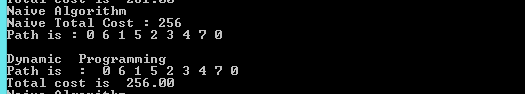
73.88,61.03,41.05,35.69,0.00,39.46,84.58,45.45

55.97,23.02,8.25,13.60,39.46,0.00,47.80,42.20

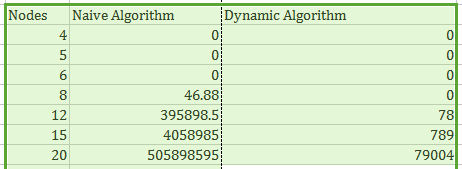
84.40,24.84,51.62,59.40,84.58,47.80,0.00,84.88

30.46,61.29,35.11,28.64,45.45,42.20,84.88,0.00

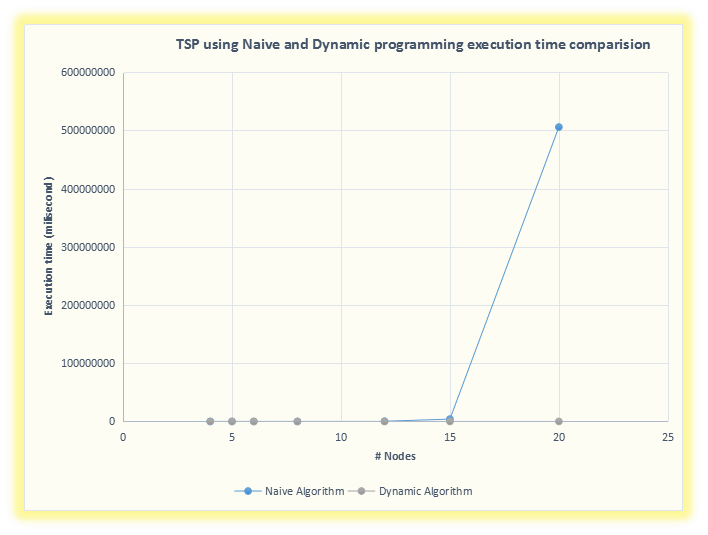
Output:



**Execution time of Naive and Dynamic Programming Algorithm:**

**Execution time in milliseconds** 

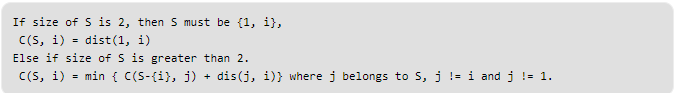
**Graph Analysis of both Algorithm:**

****

1 . From above graph and generated data from both algorithm shows Dynamic programming Algorithm perform much better than naive Algorithm but Dynamic Programming algorithm takes Extra space so Space Complexity is greater than Naive Algorithm.

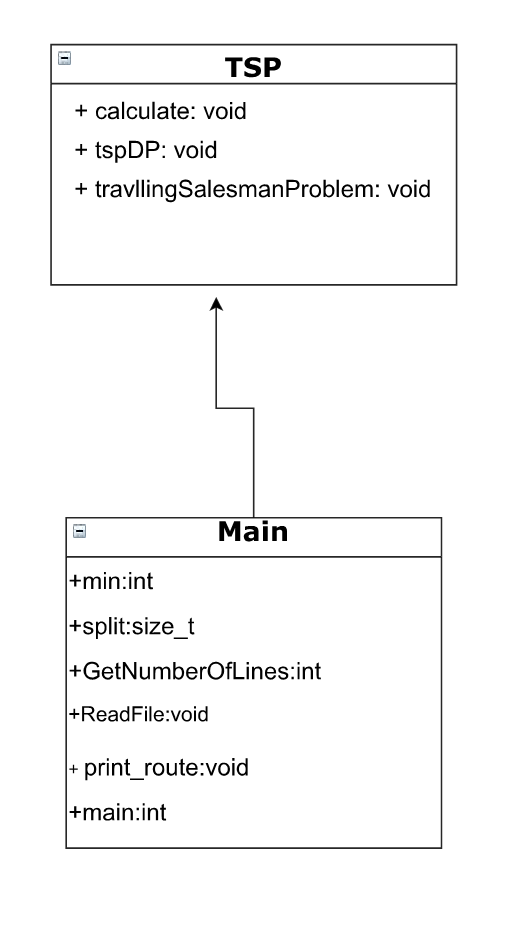
1. Dynamic programming Time complexity is better than Naive Algorithm because DP Algorithm stores previous step the computation which uses in future steps.
2. Asymptotic Time complexity of Naive Algorithm is  **O(n!)**.
3. Asymptotic Time complexity of DP Algorithm is  **O(n2\*2n)**.
4. Recurrence relation or sub problem divide in DP :

S set of vertices



**Design Pattern:** I used Behavioural design patterns for Travelling salesman problem solution. These design patterns are all about Class's objects communication. Behavioural patterns are those patterns that are most specifically concerned with communication between objects.in my case TSP class object called by main class. in TSP class there are two function which calculate Travelling salesman cities path using two different algorithm. this pattern is extendable to any level in which we can more algorithm to TSP class but drawback is when added more algorithm to TSP class make class more complex so may be user confuse.

**UML DIAGRAM:**

****

**Dynamic Programming:**

Dynamic programming is making the program efficient by storing the results. It is breaking the problem into multiple sub problems and solving each problem just once and storing the solution in a data structure. The sub problem is indexed depending on the value of the position. So, if the problem occurs again it will not have to repeat yourself over and over again, you can just look at the sub problem solution.

**Conclusion:**

Travelling Salesman problem solve through Dynamic Programming is better than Naive Algorithm because Naive Algorithm Execution time increase much larger than DP with Exponentially.