Travelling Salesman Problem Solved Using Simulated Annealing

Prof. Pratik Shah

Newbies@AI

Kaishav Basodiya	202051094
Amanshu jaiswal	202051023
Abhishek Kumar	202051003
Aditya Priyanshu	202051010

What is Travelling Salesman Problem-

Given a set of cities and distances between each pair of cities, what is the shortest possible route that visits each city exactly once before returning to origin city.

Why TSP is NP-Hard Problem -

it involves finding the shortest possible route that visits a given set of cities and returns to the starting city. The problem becomes more difficult as the number of cities increases, and it is not known whether a polynomial-time algorithm for solving the problem exists.The TSP belongs to a class of problems called combinatorial optimization problems, where the goal is to find the best solution from a finite set of possibilities. TSP is hard because the number of possible routes increases exponentially with the number of cities, making it impossible to check all possible routes.

Constraints For TSP -

- Graph should be complete graph.
- Edge Weights must be positive.
- Every Node must be visited only once.
- Salesman must reach same destination as origin.
- Unknown/Missing Edge weights are considered to be infinite.

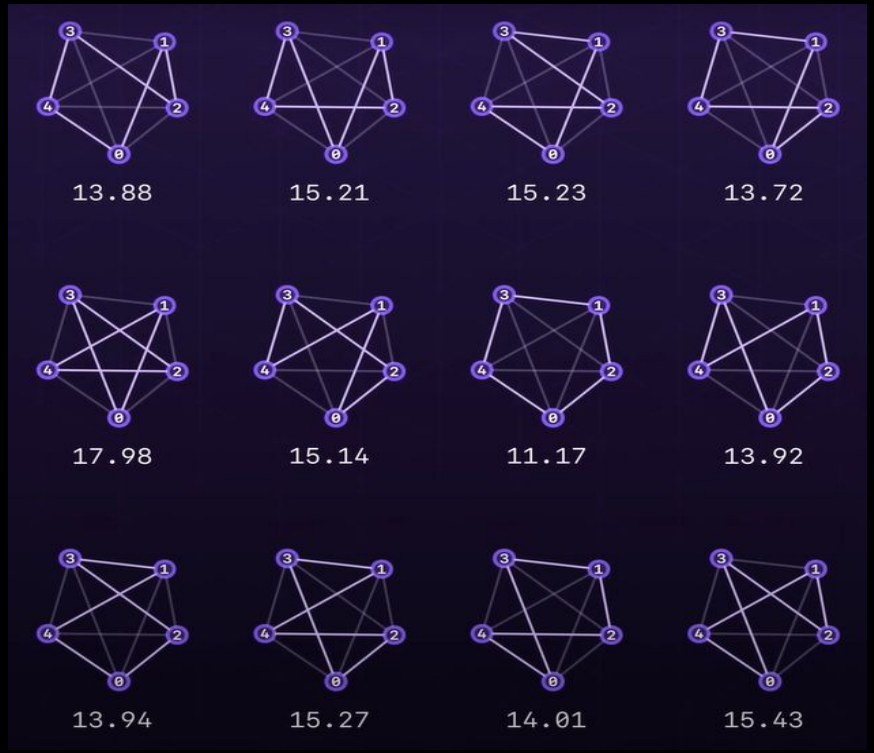
Approaches For TSP -

- Brute Force
- Nearest neighbour heuristic
- Minimum Spanning Tree
- Christofides Algorithm
- Simulated Annealing

Brute Force

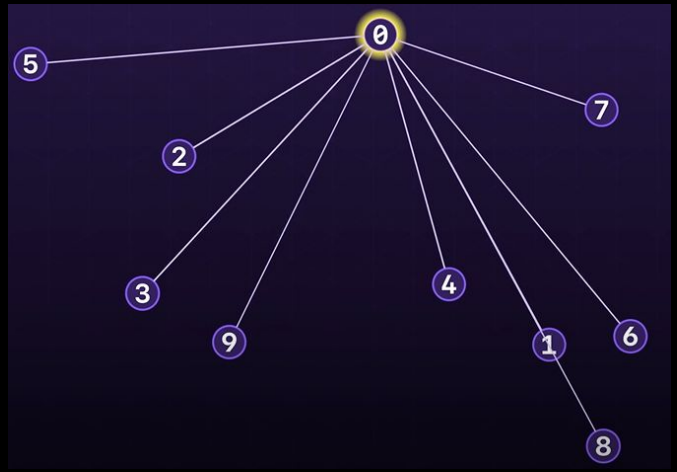
To Explore all possible tours and then taking the minimum cost tour out of all of them.

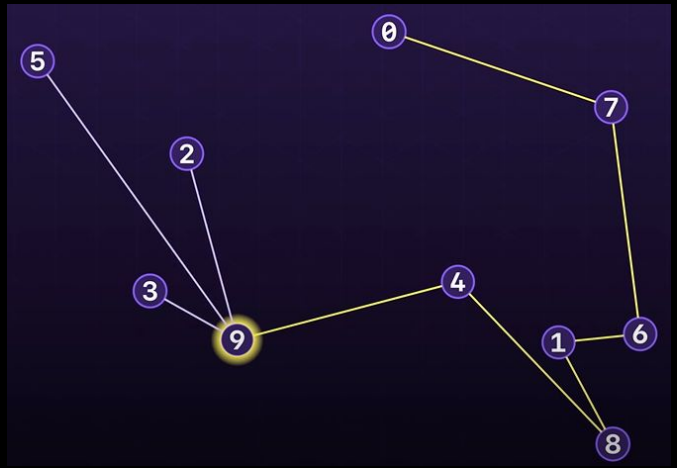
Time Complexity - O((n-1)!/2)



Nearest neighbour heuristic

It is a greedy approach. It starts on a node then visits the nearest node to current node move to nearest neighbour as long as it doesn't close loop prematurely.





Minimum Spanning Tree

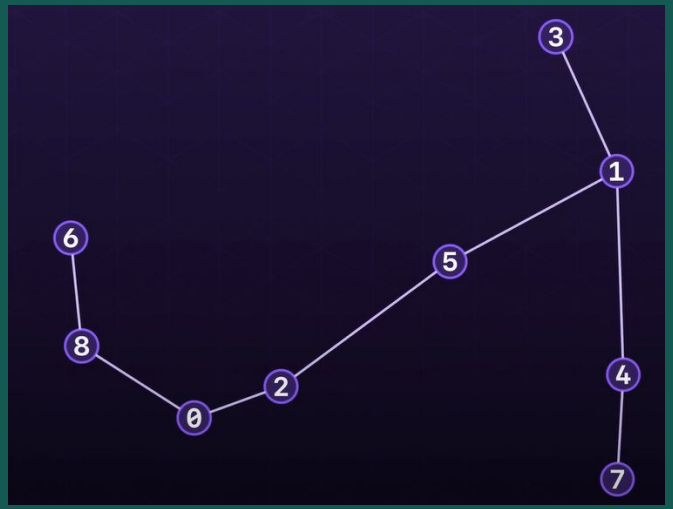
(To set the lower bound of optimal solution of TSP for large dataset)

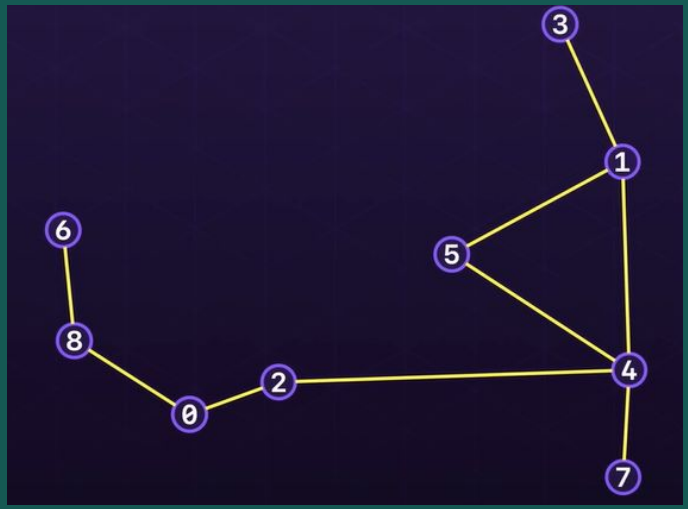
It is a subgraph of a undirected connected graph which includes all vertices connected with minimum no. of edges with lowest sum of edge weights.

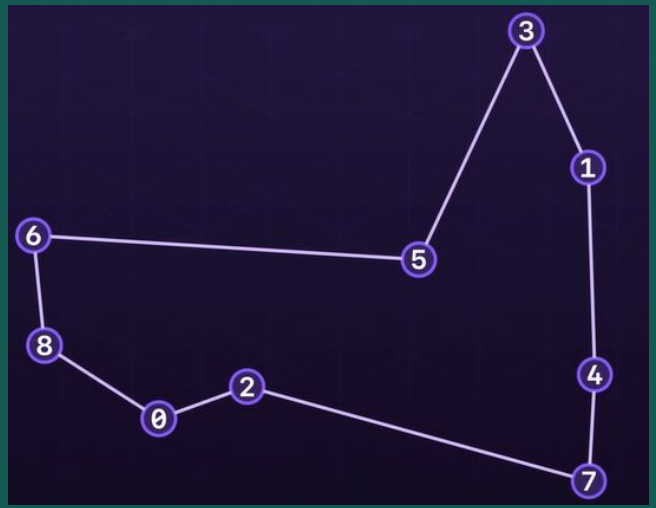
MST can be considered as lower bound for TSP because we don't know the optimal solution of TSP.

Prim's Algorithm is a way to find Minimum Spanning tree.

One Tree with maximum edge weight sum is closest to optimal solution of TSP.


3
1
6
5
8
4
2
0
7

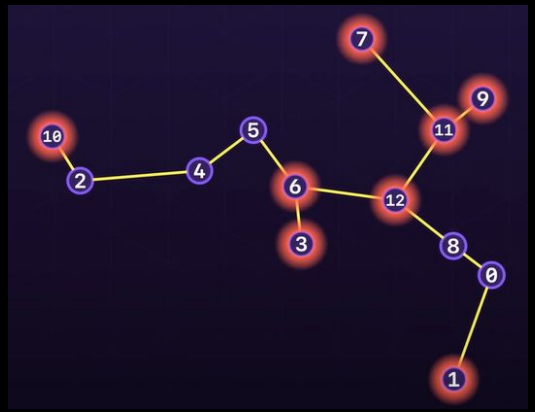

3
1
6
5
8
4
2
0
7

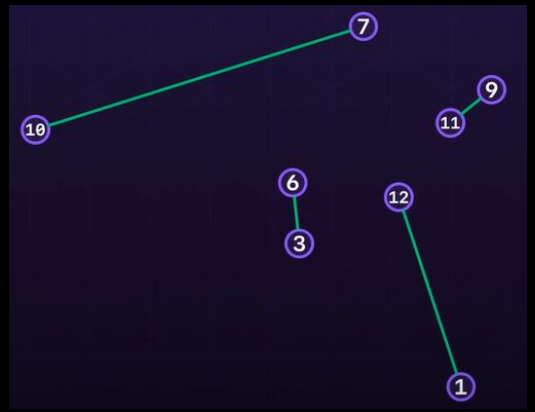

3
1
6
5
8
4
2
0
7

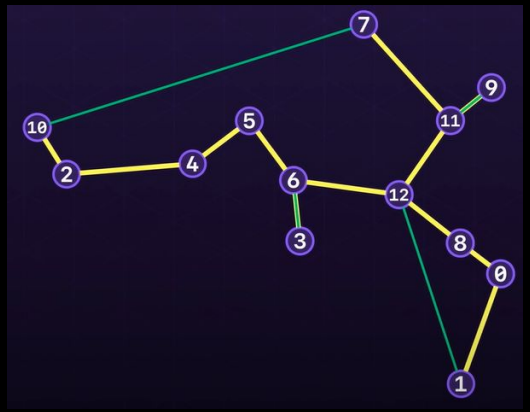
Christofides Algorithm

It is based on getting Optimal Solution from the MST.

- In this algo, we isolate the odd degree vertices ‘S’ .
- try to find min weight perfect matching Tree ‘M’ of ‘S’.
- Combine M and MST into a Multi-graph
- Generate Eulerian Tour Of Multi-graph
- Get TSP tour from Eulerian Tour.





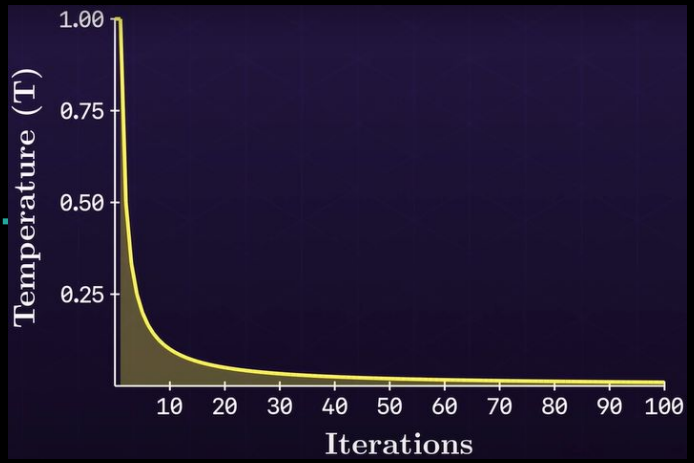


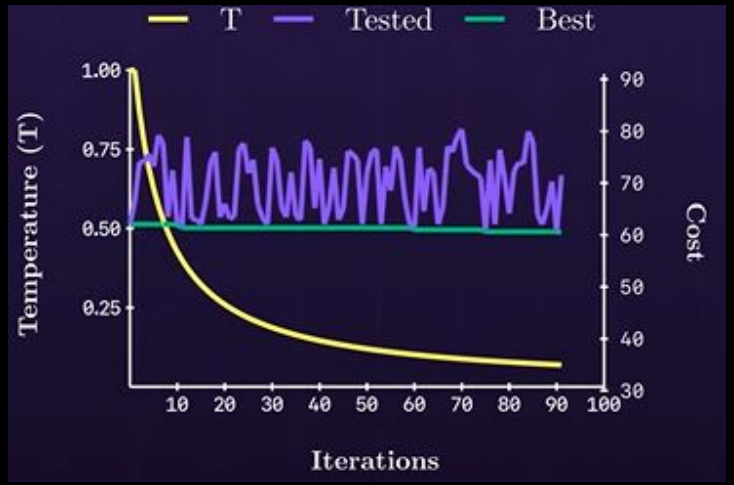
Tour Improvements(Local Search)

- Random swapping - randomly pick two nodes in tour order and swap them, if cost of tour reduces then accept it else reject it.
- 2-opt Optimization - there are some pairs of edges which if we connected in only alternative valid order improves the tour.
- K-opt Optimization - it is same as 2-opt but we take k edges into consideration.

Simulated Annealing

- Main idea of this approach is to even accept the worse solution in the hope of getting a better overall solution.
- We start with a candidate solution and then apply tour improvements using random swapping/2-opt/3-opt





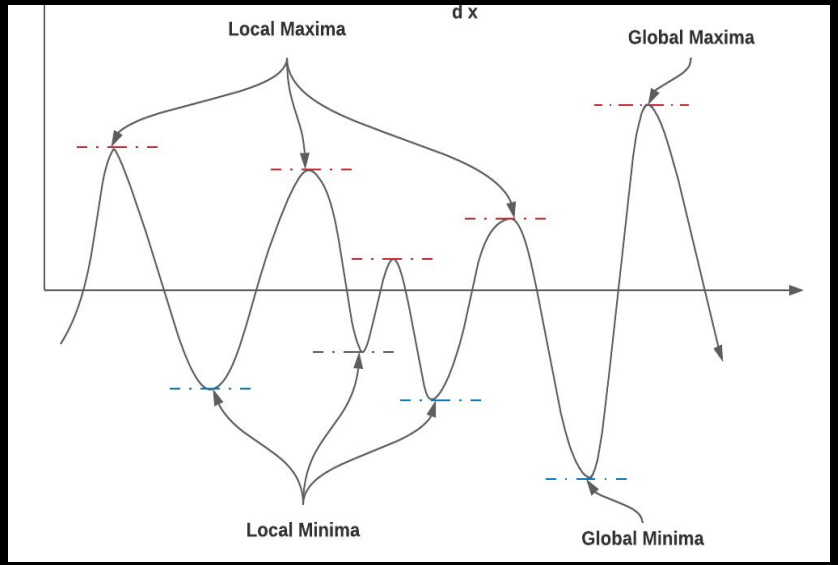
Simulated Annealing Working

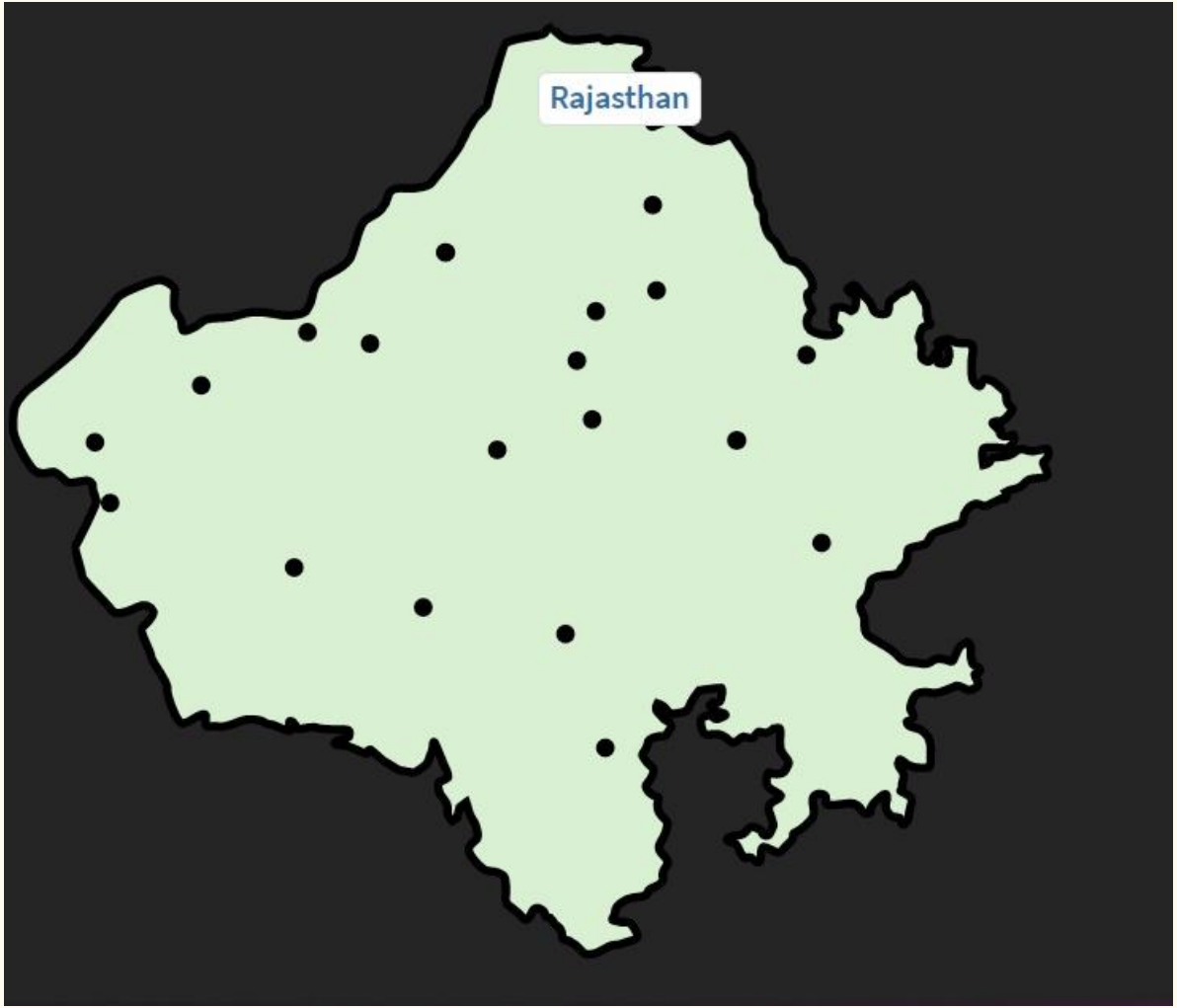
Generate an initial candidate Solution(Tour) randomly.
Set starting Temperature(T0), cooling factor(α).
Randomly swap two nodes in tour.
Check if Current Tour is better than previous.
If better -> accept it for next iteration
else ->accept if (random(0,1)< probability 1/(1+ exp((c1-c2)/T))).
Check for stop condition.
Decrease the temperature (T = T0*α).
Reiterate for the current tour.

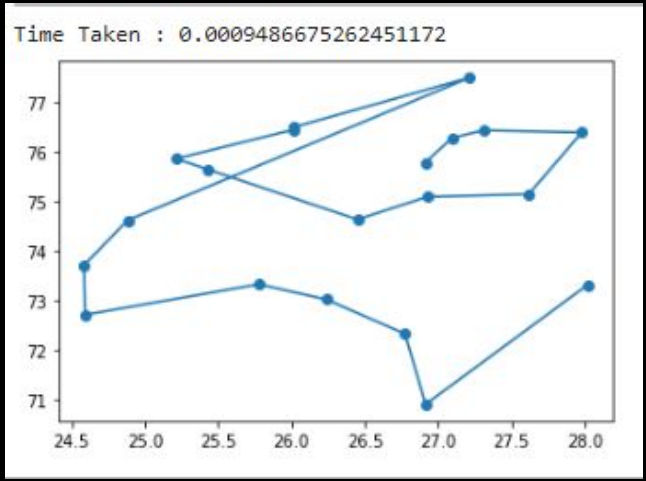
Simulated Annealing

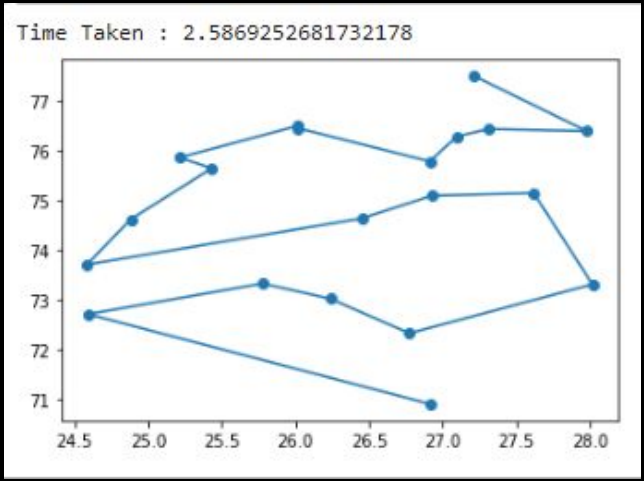
Various Factors Affecting SA -

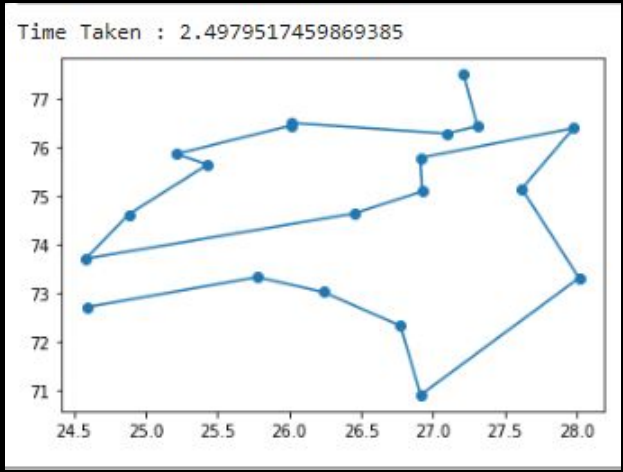
- Cooling schedule
- Number of Cities
- Randomness of picking initial path

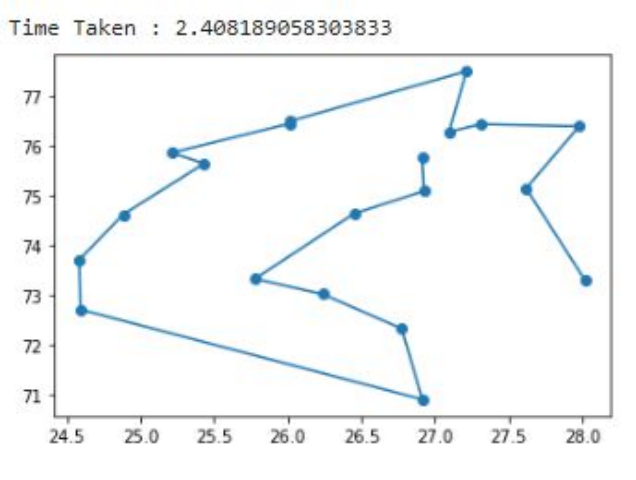



Rajasthan

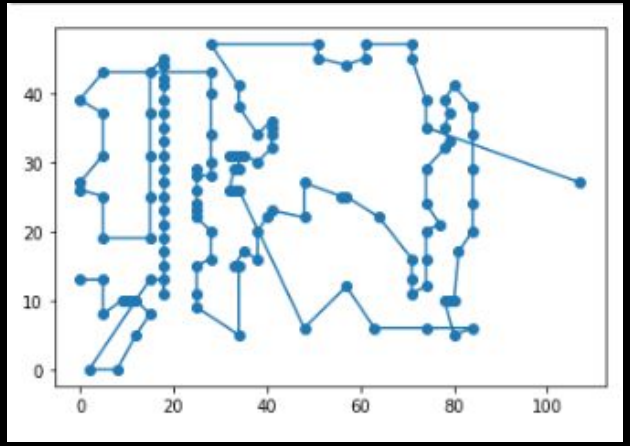

Time Taken : 0.0009486675262451172
77
76
75
74
73
72
71
24.5
25.0
25.5
26.0
26.5
27.0
27.5
28.0

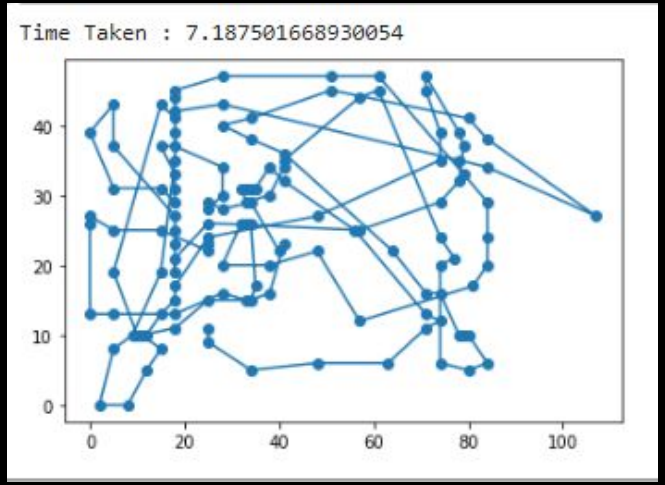

Time Taken : 2.5869252681732178
77
76
75
74
73
72
71
24.5
25.0
25.5
26.0
26.5
27.0
27.5
28.0

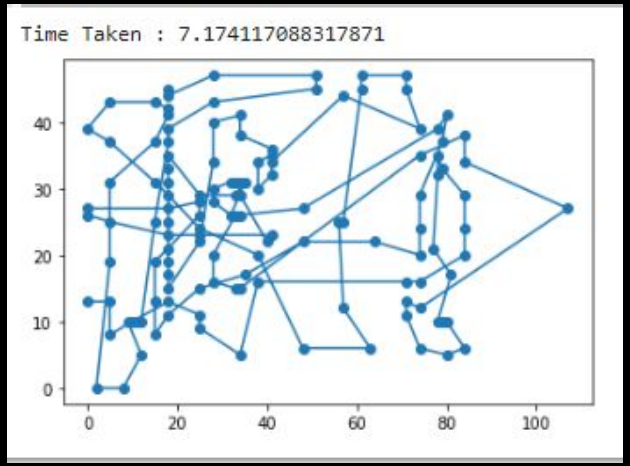

Time Taken : 2.4979517459869385
77
76
75
74
73
72
71
24.5
25.0
25.5
26.0
26.5
27.0
27.5
28.0

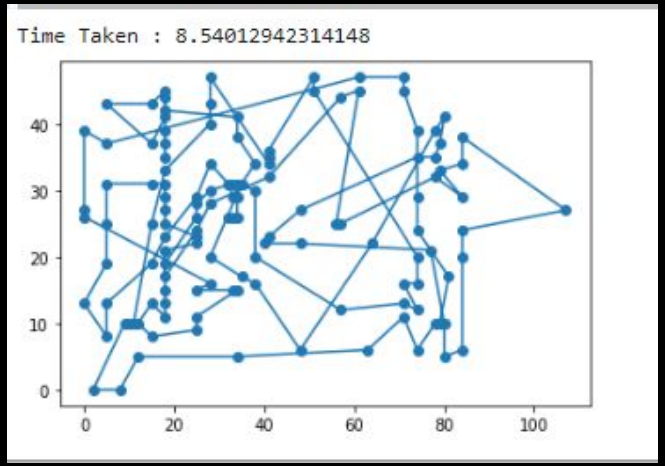

Time Taken : 2.408189058303833
77
76
75
74
73
72
71
24.5
25.0
25.5
26.0
26.5
27.0
27.5
28.0

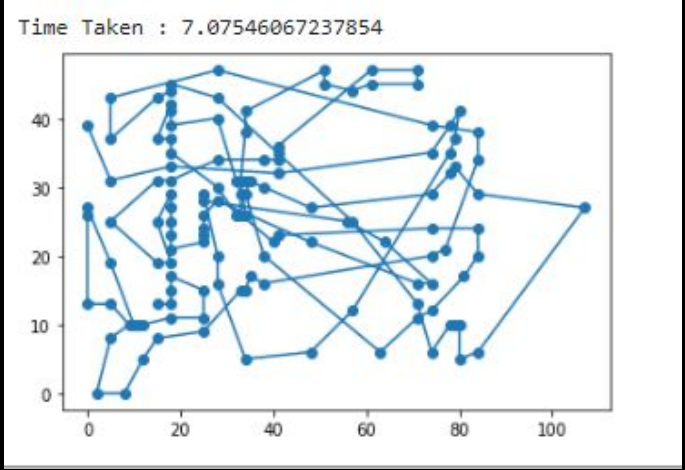
Data Set Of 131 Points

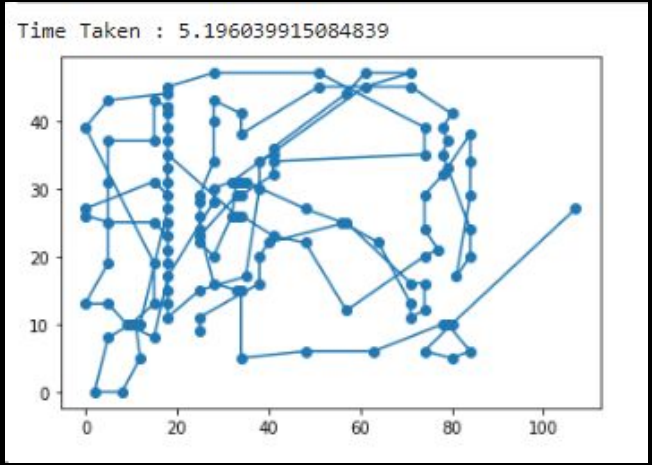


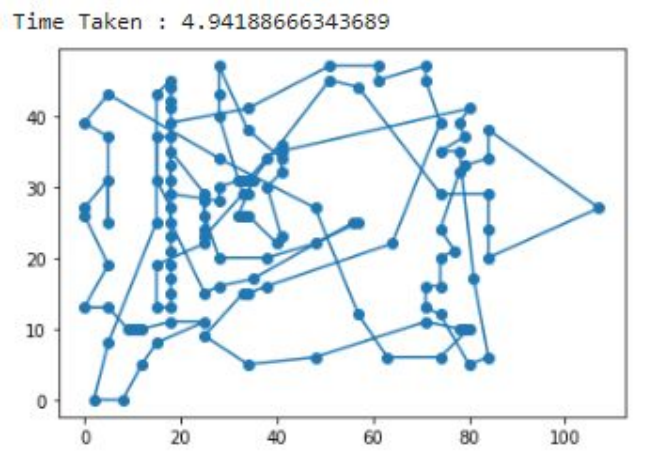


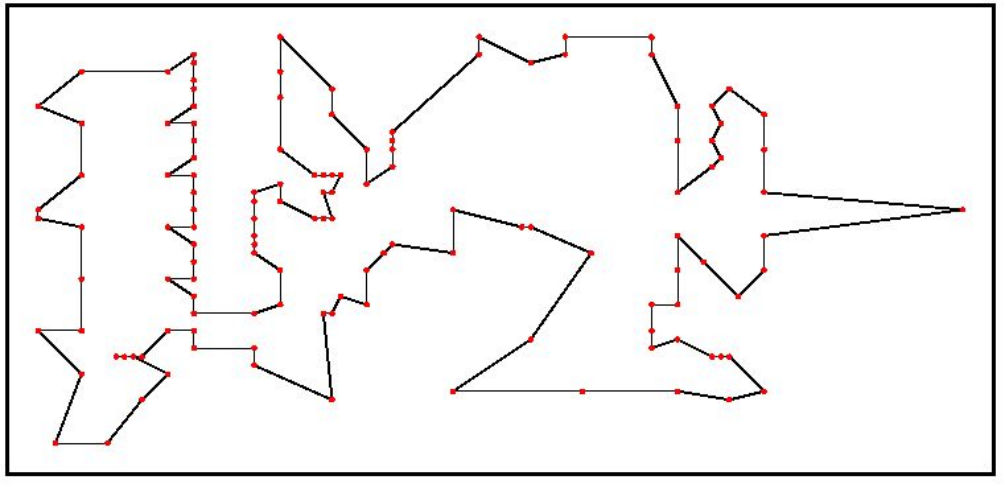




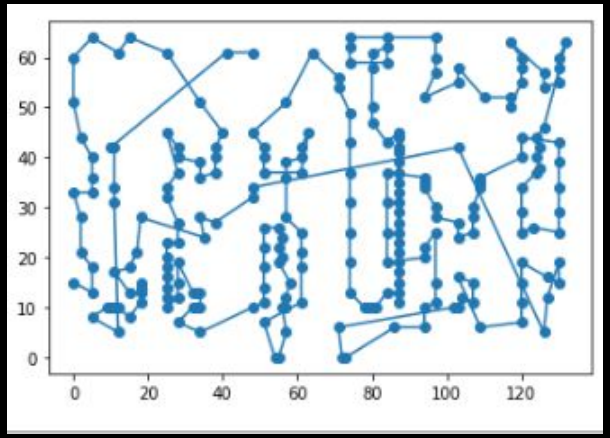

Time Taken : 7.07546067237854
0
10
20
30
40
0
20
40
60
80
100

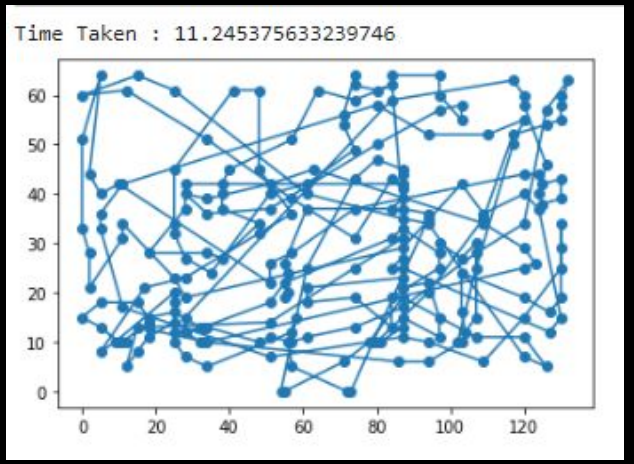

Time Taken : 5.196039915084839
0
10
20
30
40
0
20
40
60
80
100

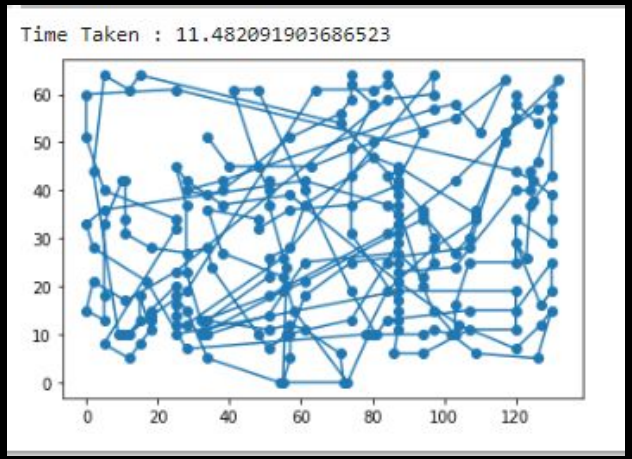

Time Taken : 4.94188666343689
0
10
20
30
40
0
20
40
60
80
100

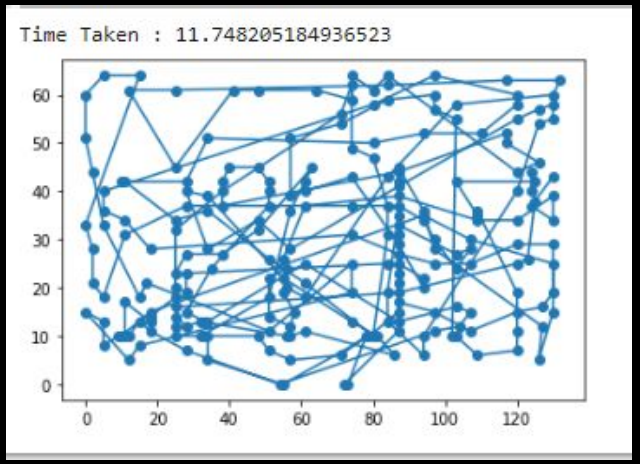


Data Set Of 237 Points

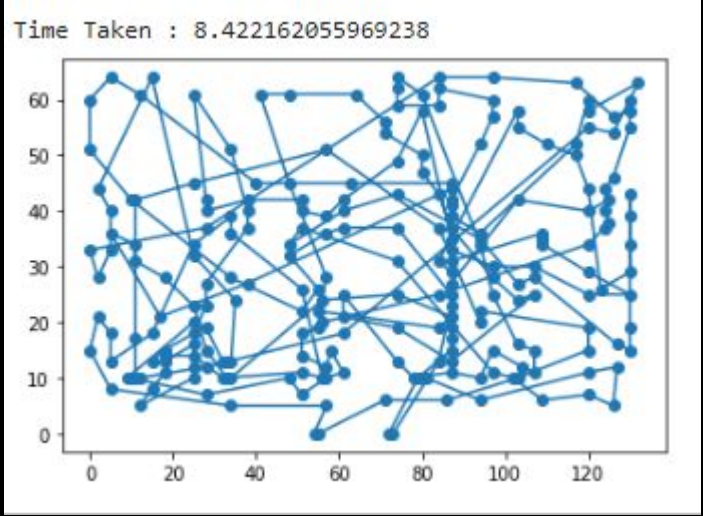


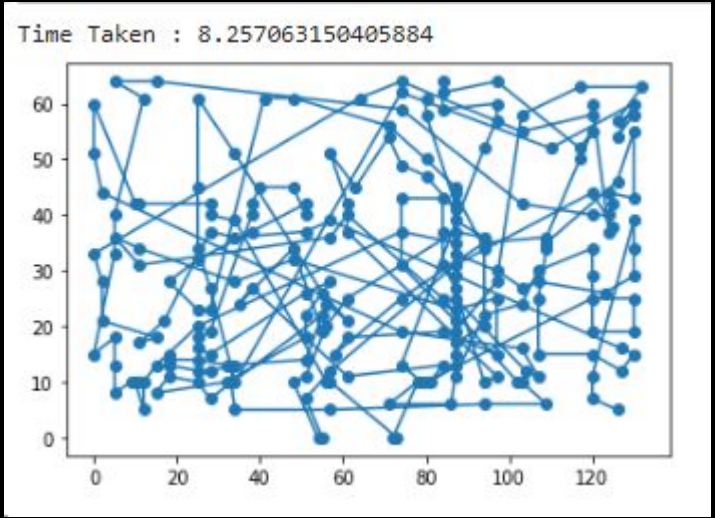


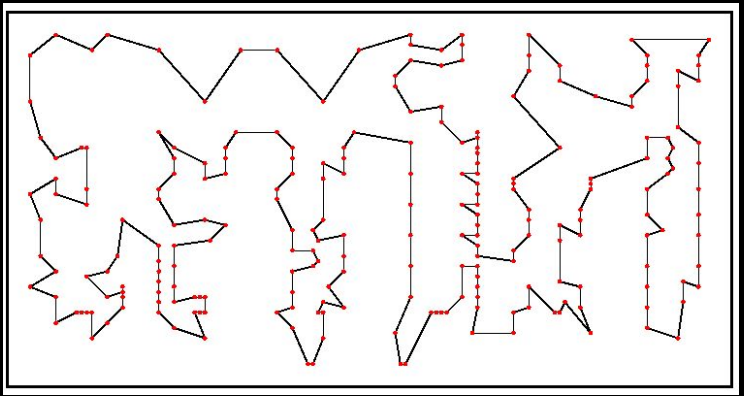




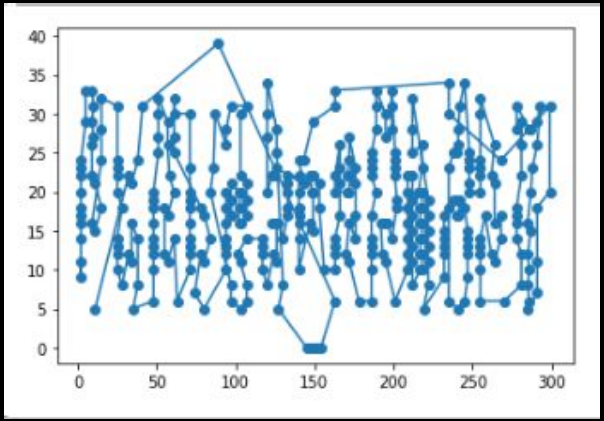
Data Set Of 237 Points

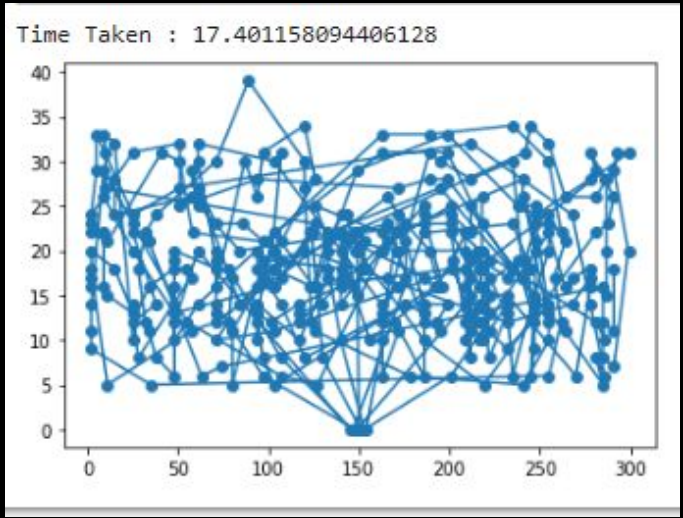


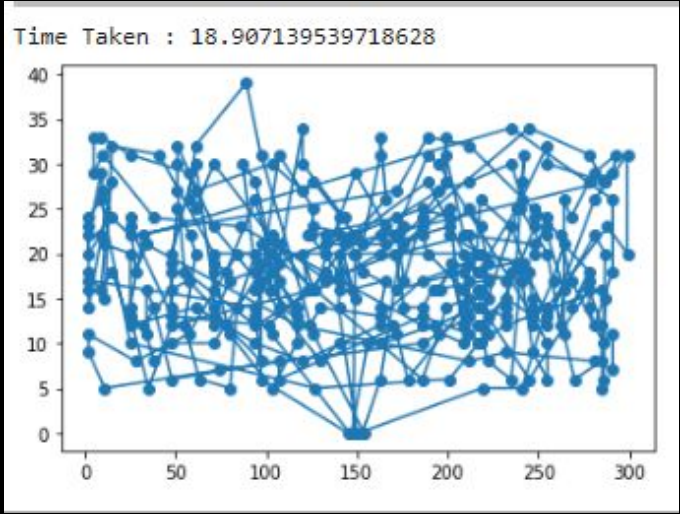


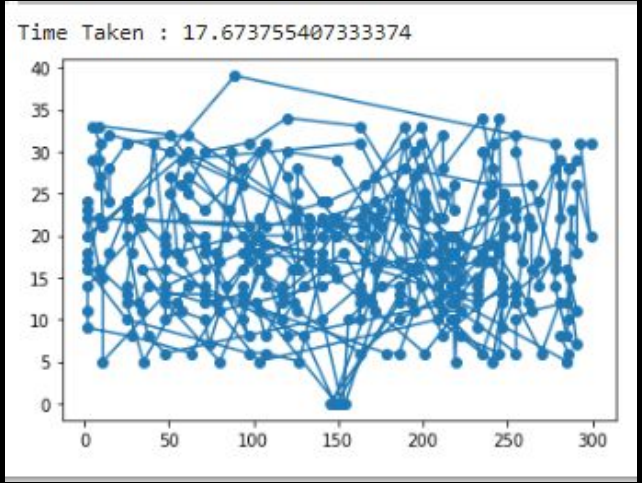


Data Set Of 343 Points

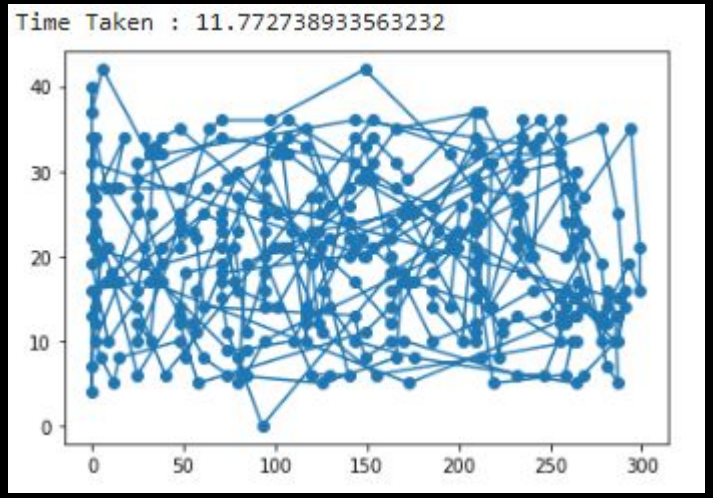


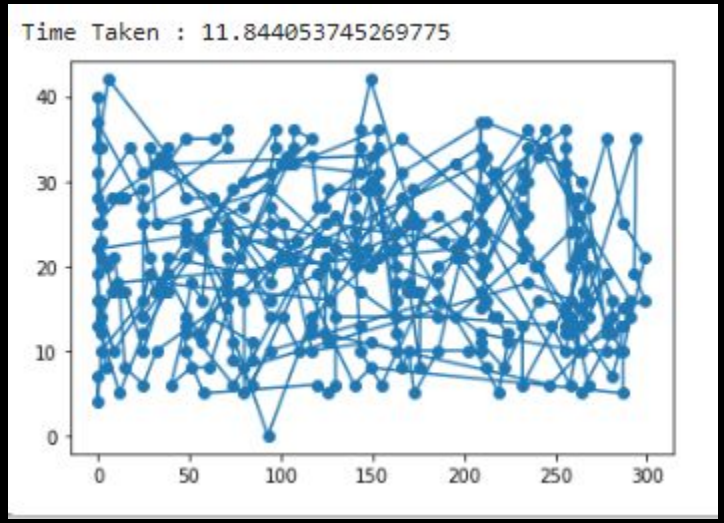


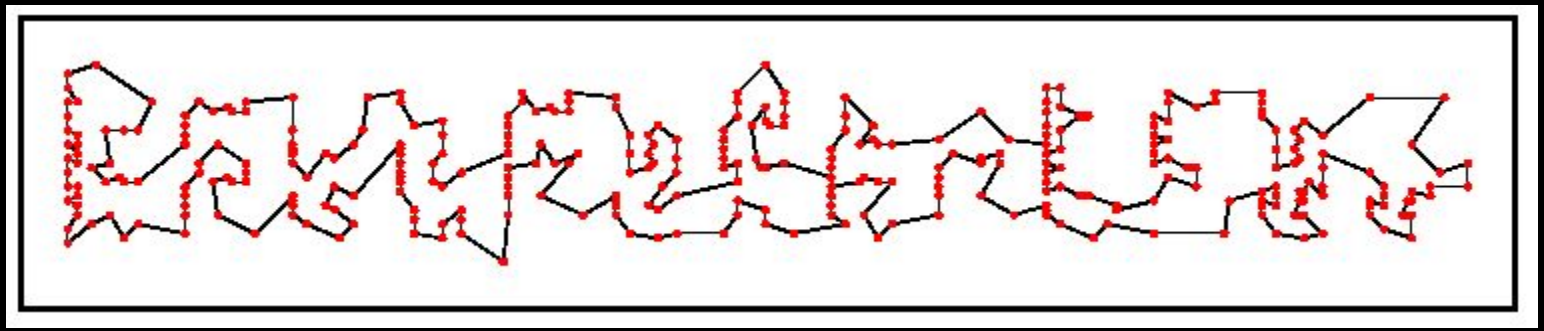




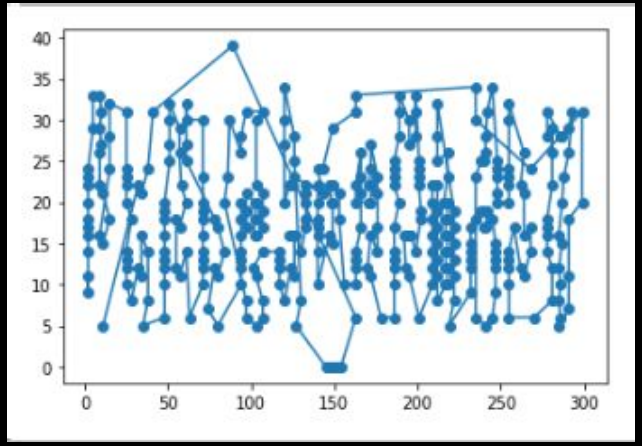
Data Set Of 343 Points

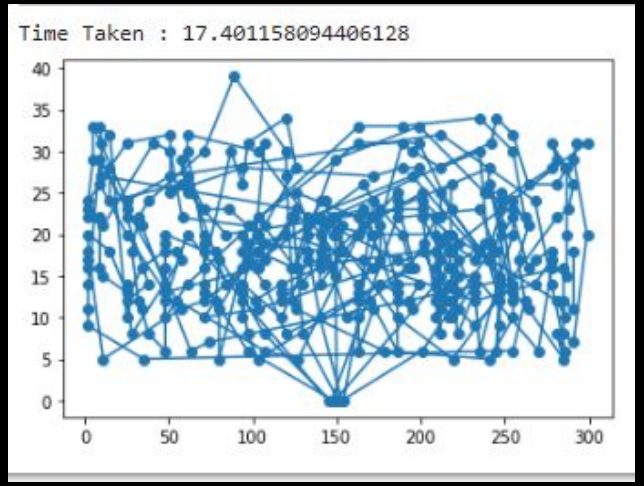


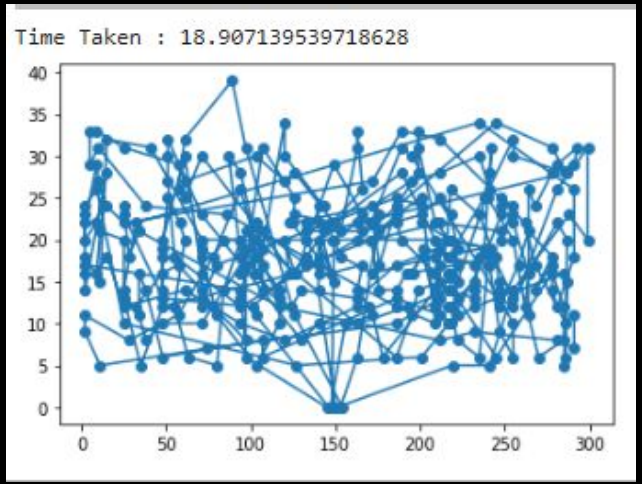


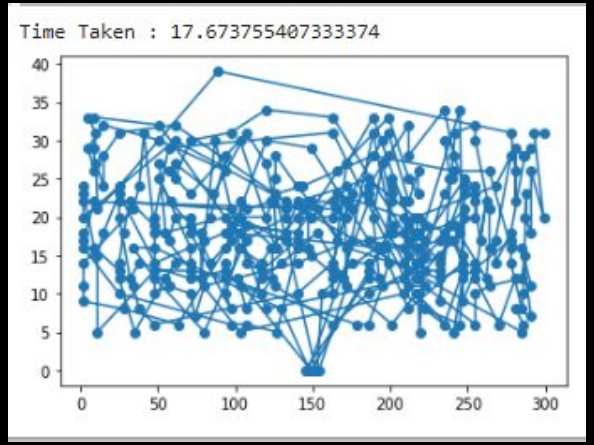


Data Set Of 379 Points

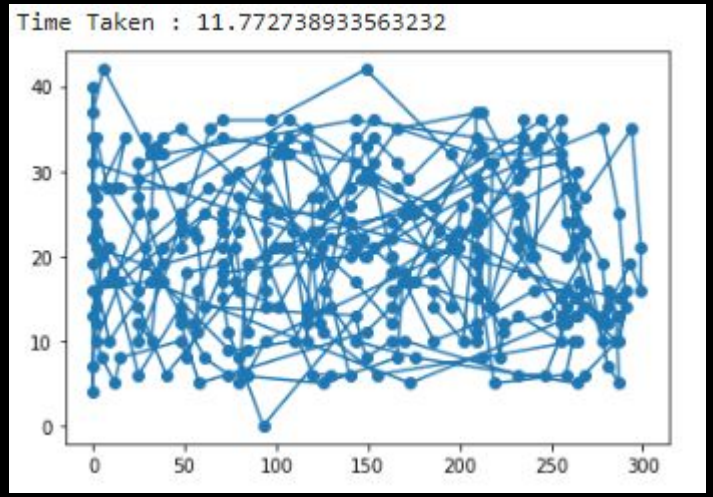


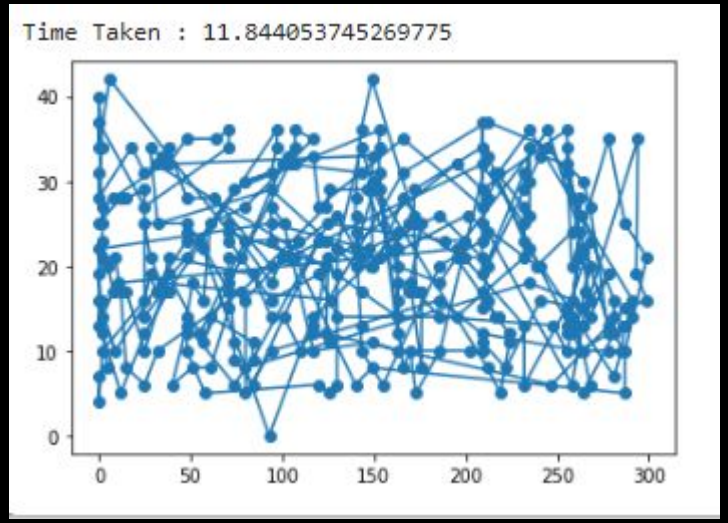


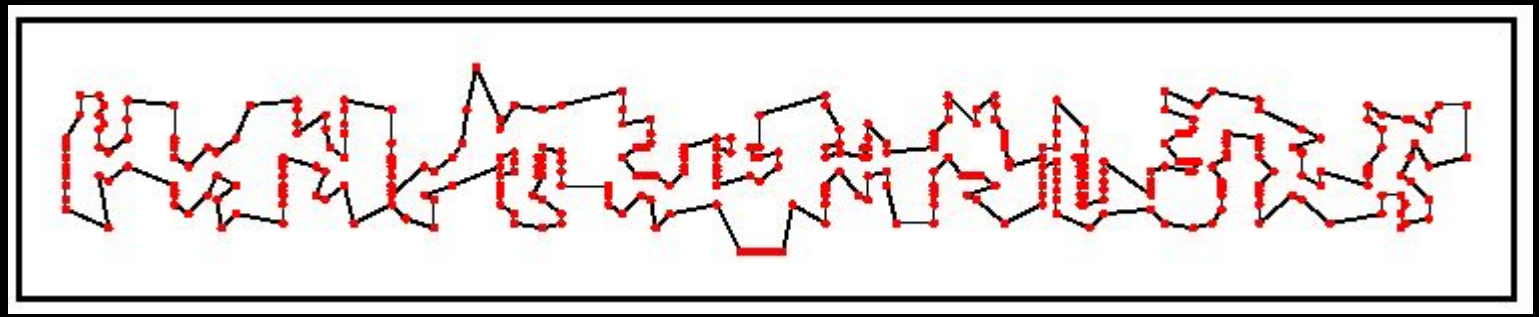




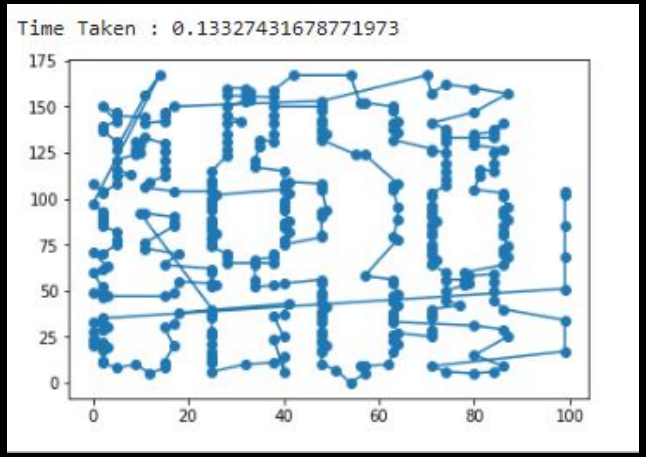
Data Set Of 379 Points

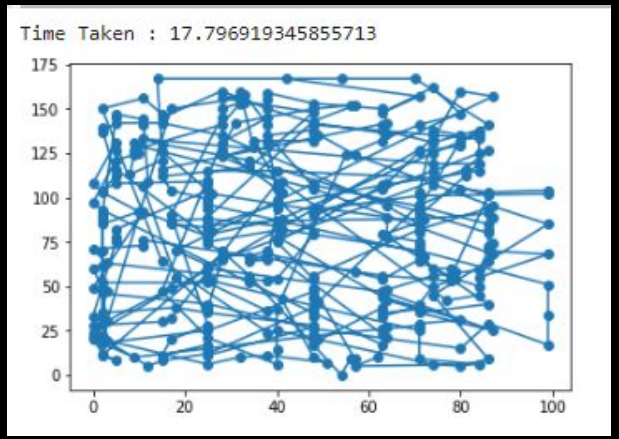


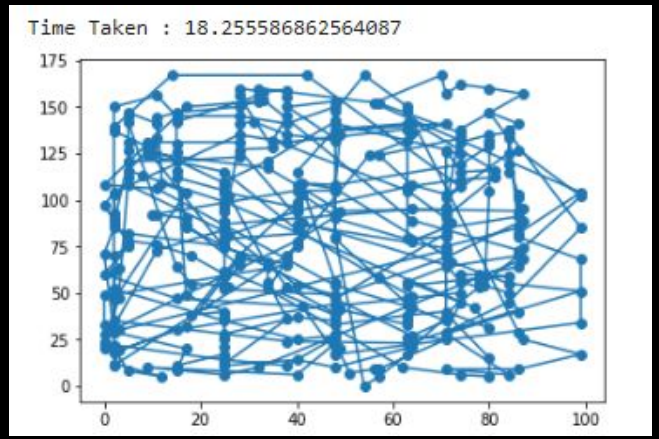


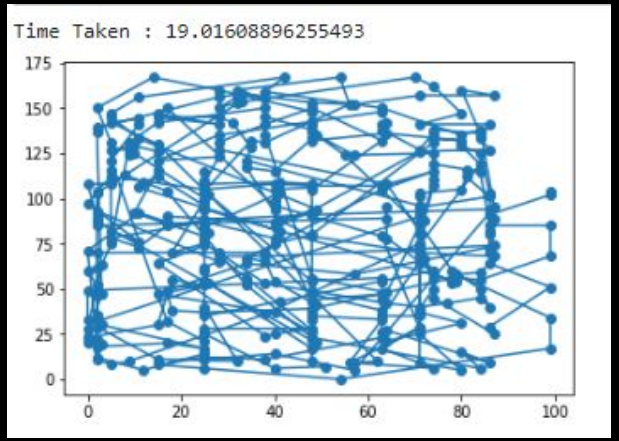


Data Set Of 380 Points

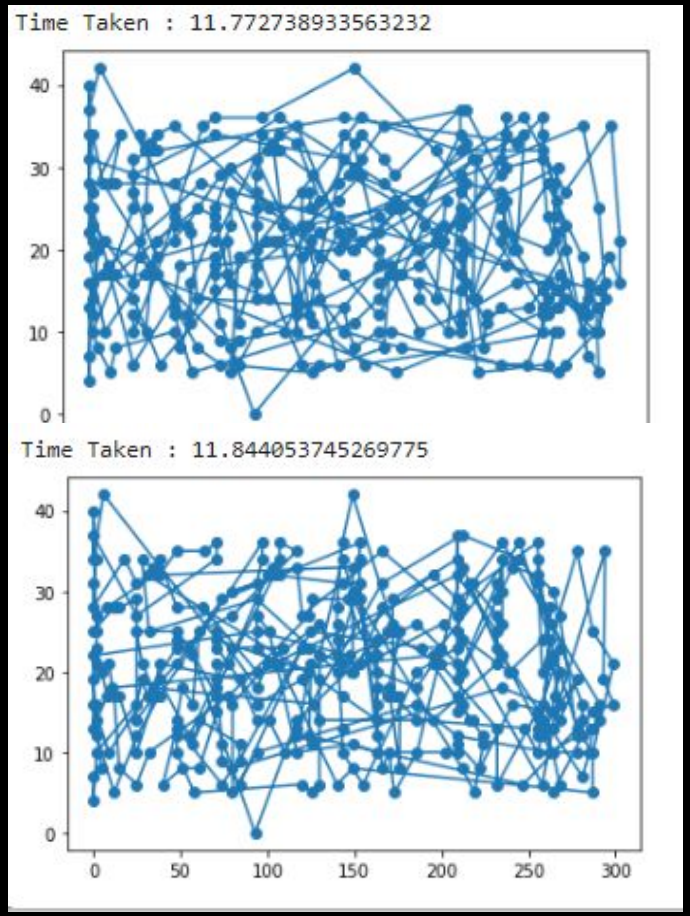


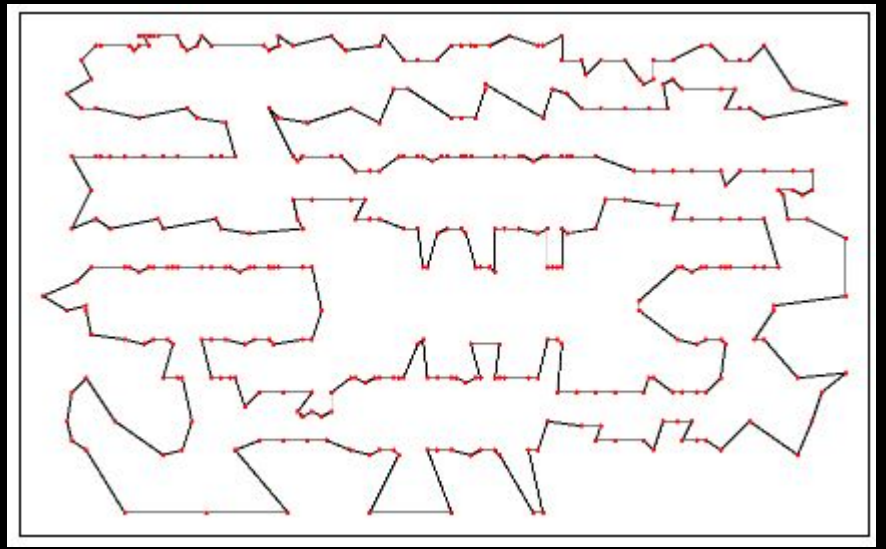






Data Set Of 380 Points





References-

A first Course in Artificial Intelligence by Dr. D Khemani
https://docs.python.org/3/
https://docs.scipy.org/doc/scipy/
https://numpy.org/doc/
https://en.wikipedia.org/wiki/Travelling_salesman_problem
https://en.wikipedia.org/wiki/Simulated_annealing