Problem Statement

Given a set of cities and distance between every pair of cities as an adjacency matrix, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point.

It is a NP Hard Problem.

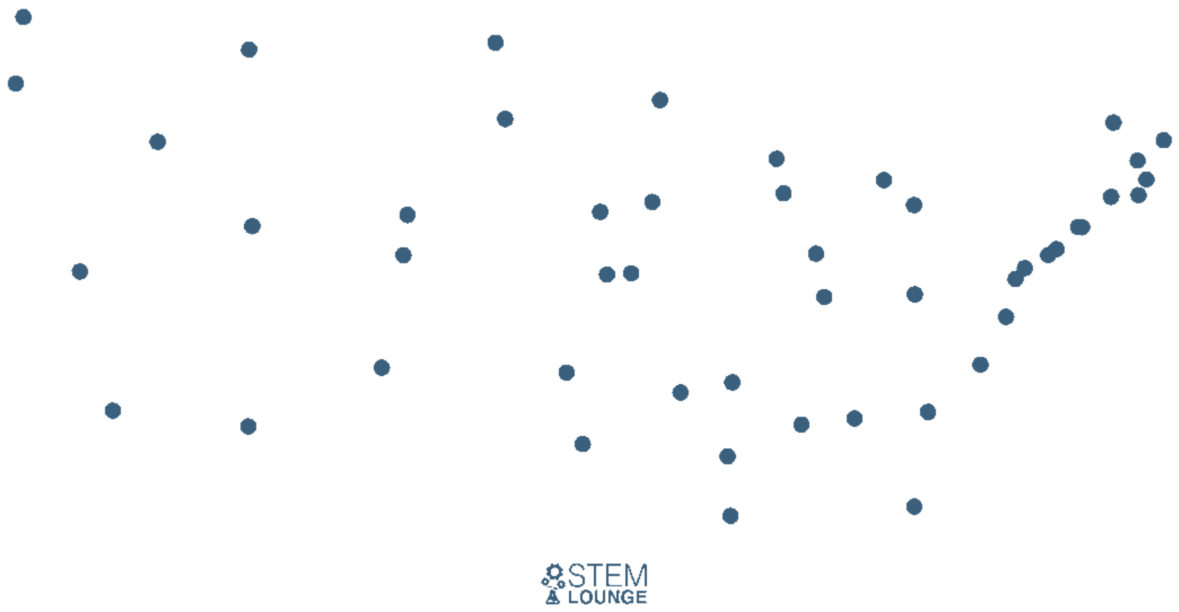
There are several ways to solve this problem

- Nearest Neighbour Heuristic
- Greedy Heuristic
- Saving Heuristic

We have solved this problem using Nearest Neighbour Heuristic.

Nearest Neighbour

The nearest neighbor heuristic is another greedy algorithm, or what some may call naive. It starts at one city and connects with the closest unvisited city. It repeats until every city has been visited. It then returns to the starting city.



Nearest Neighbor Heuristic

Karl Menger, who first defined the TSP, noted that nearest neighbor is a sub-optimal method:
"*The rule that one first should go from the starting point to the closest point, then to the point closest to this, etc., in general does not yield the shortest route."*

The time complexity of the nearest neighbor algorithm is `O(n^2)`. The number of computations required will not grow faster than n^2.

Here we use Greedy approach. we first consider a random city and
moves to the nearest neighbor keeping in mind we don't form a premature loop.

Iterative Improvement

for the best path we choose all the nodes as the starting node one by one and whichever path gives us the best path with least cost we print that one.