Traveling Salesperson Problem

1. Introduction

Traveling Salesperson Problem: TSP is a problem that tries to find a tour of minimum cost that visits every city once. In this visualization, it is assumed that the underlying graph is a complete graph with (near-)metric distance (meaning the distance function satisfies the triangle inequality) by taking the distance of two points and round it to the nearest integer.

2. Visualization

View the visualisation of TSP algorithm here.

Originally, all edges in the input graph are colored with the grey.

Throughout the visualization, traversed edge will be highlighted with orange.

3. Input

There are two different sources for specifying an input graph:

1. **Draw Graph**: You can put several points on the drawing box, but you must not draw any edge to ensure the (near-)metric property of the graph. After you have finished putting the points, the edges will be drawn automatically for you after.
2. **Example Graphs**: You can select from the list of graphs to get you started.

4. Bruteforce

Bruteforce: It tries all (V-1)! permutation of vertices (not all V! since it does not matter where we start from). It enumerates all possibilities by doing a dfs search with parameters similar to those of Held-Karp algorithm, that is, the DFS search will return the value of the tour starting from current vertex to all vertices that we have not already visited.

Time complexity: O(V * (V - 1)!) = O(V!).

5. Dynamic Programming

Dynamic Programming: It uses a widely known algorithm called Held-Karp. In this visualization, it is implemented as a DFS search that is the same with the bruteforce algorithm, but with memoization to cache the answers. This dramatically brings down the run time complexity to O(2^V * V^2).

Time complexity: O(2^V * V^2).

Note that for N = 10, this algorithm takes roughly ~(100 * 2^10) = 102K operations while the bruteforce algorithm takes roughly ~(10!) = 3628800 operations, around 30 times faster.

6. Approximation

Approximation: There are two approximation algorithms available, a 2-approximation algorithm and a 1.5-approximation algorithm that is also well known as Christofides Algorithm.