Tecnológico de Monterrey

Algorithm Explanation

28/11/2023

Armando Terrazas Gómez | A01640924

Analysis and Design of Advanced Algorithms

Gildardo Sanchez-Ante

Algorithm Selection

Kruskal's Algorithm for Connecting Points

We used Kruskal's algorithm to connect all the points in the best way possible, making sure we didn't go over any path twice. It sorts all the connections from shortest to longest and then picks them one by one, but it skips any that would make a loop. This method is quick, taking about $O(E \log E)$time, where E is how many connections we have. This is much faster than a basic way, which would check every possible way to connect the points.

Traveling Salesman Problem (TSP)

For the Traveling Salesman Problem, where you need to find the shortest way to visit all the points and come back, in our case we used dynamic programming. The simple way would explore every possible route, leading to a time complexity of $O(n!)$, where n is the number of points. The dynamic programming solution, however, reduces this to $O(n^2 \cdot 2^n)$, a substantial improvement for smaller datasets.

Edmonds-Karp for Maximal Flow Through a Network

Then, we used the Edmonds-Karp algorithm to figure out the best way to send stuff through the network. It keeps looking for the best path to send more stuff until it can't find any more. This way takes $O(V \cdot E^2)$ time, where V is how many junctions and E is how many pipes we have. The simple way, which would try every possible way to send stuff, and the complexity rises a lot.

Making a Voronoi Diagram

Lastly, we made a Voronoi diagram. We did this by first computing the Delaunay Triangulation between all the points and then finding the circumcenter of these triangles. This method is faster than the naive approach and takes about $O(n \log n)$ time, where n is the number of points.

What I Learned?

This project taught me how picking the right way to solve a problem can make a big difference. It's not just about finding an answer, but finding the best and fastest answer. These smart methods like Kruskal's for connecting points, dynamic programming for the traveling salesman, Edmonds-Karp for network flows, and making Voronoi diagrams for geometry are really useful. They save a lot of time and help me think of better ways to solve problems, not just the first way that comes to mind. This is really helpful for my coding projects and in competitions where being quick and efficient is really important.