Euclidean Minimum Spanning Tree

Problem Statement

The Euclidean minimum spanning tree is a kind of MST that the distance of each two vertices is also the weight of the edge between the two vertices. The problem is to determine if the Euclidean minimum spanning tree of n points can be computed in time close to the lower bound of $\Omega(n \log n)$.

Background

According to the class website, previous work about this problem include:

1. Chazelle presented an $O(m\alpha(m, n)\log\alpha(m, n))$-time algorithm
2. An $O(m\alpha(m, n))$-time algorithm [Cha00b], where $\alpha(m, n)$ is the functional inverse of Ackermann's function, and n and m are the number of vertices and edges respectively in the graph.
3. Pettie and Ramachandran have since given an optimal algorithm for the graph setting, , whose running time is an unknown function between $\Omega(m)$ and$O(m\alpha(m, n))$.

Proposal

I plan to study the previous work done by other people. Then based on their work. I'll try to see if I can implement their algorithms and if there is a better solution.

I also found another article Efficient minimum spanning tree construction without Delaunay triangulation related to this problem. The authors of this article claim that their algorithm for this problem could solve this problem in O(n logn) time. I plan to also study this article and see if I can implement their algorithm.