**Name:** Serhat Gürgenyatağı

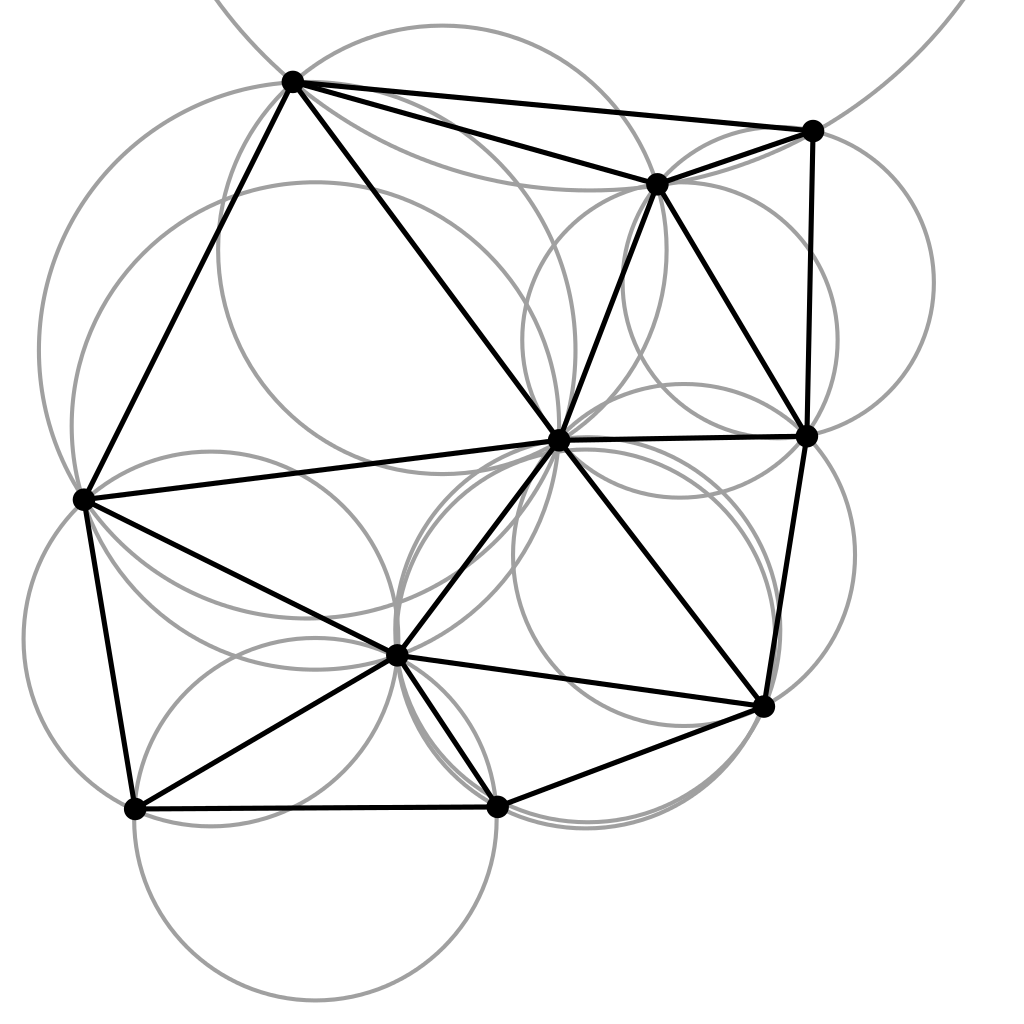
**ID:** 21903288

CS478 – Computational Geometry

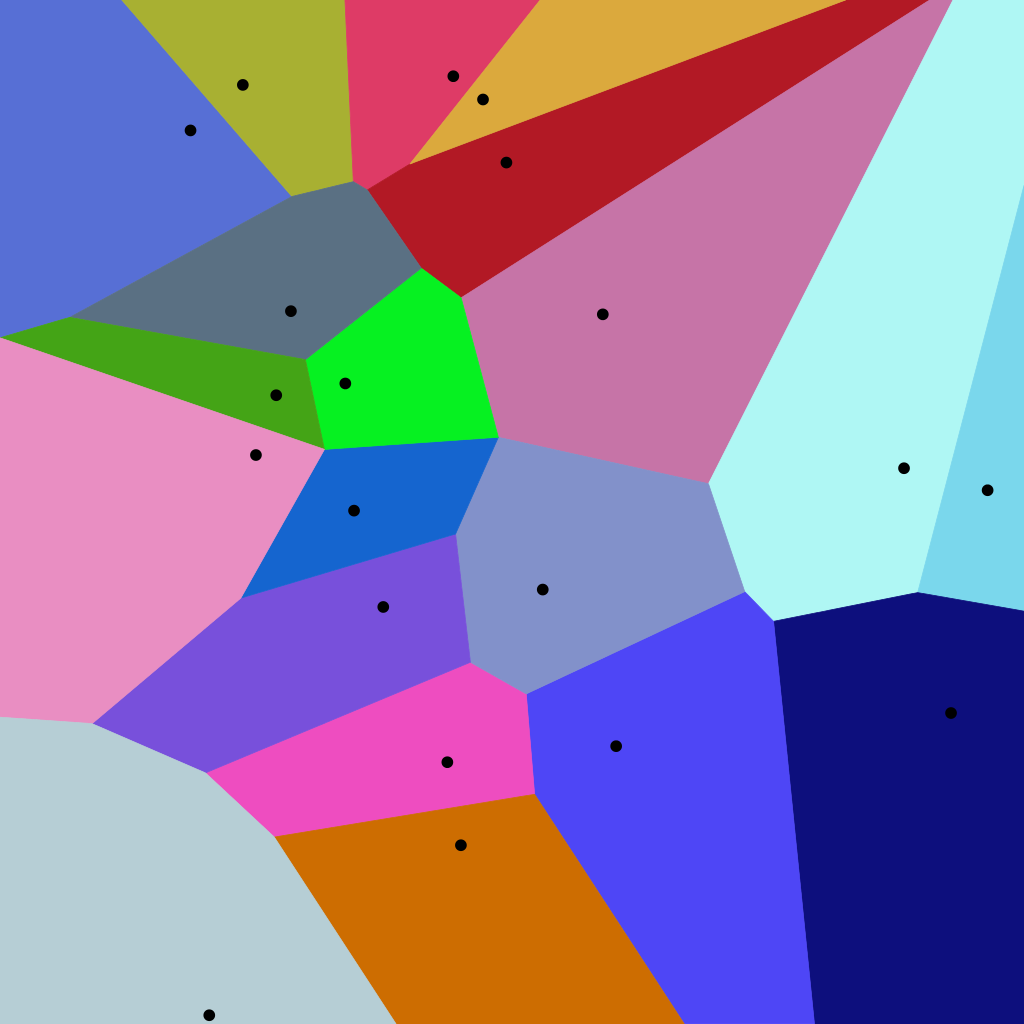
**Project Final Report**

**Survey of the Subject Area**

Delaunay Triangulation, is a triangulation of a given set of discrete points, not all colinear, such that none of the circumcircles of the triangles includes a point inside. This definition also implies that no edges can be added to the triangulation without crossing other edges.



Voronoi Diagram is the partitions of a given set of points such that each region consists of all points that are closer the center point than any other that is in the set. Delaunay Triangulation is the dual graph of Voronoi Diagram (which means that it has vertices for every face of the Voronoi Diagram and vice versa).

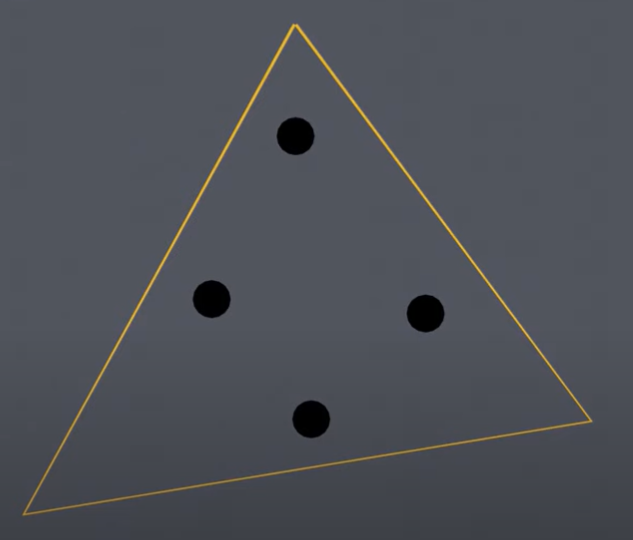


**Algorithms to be Used**

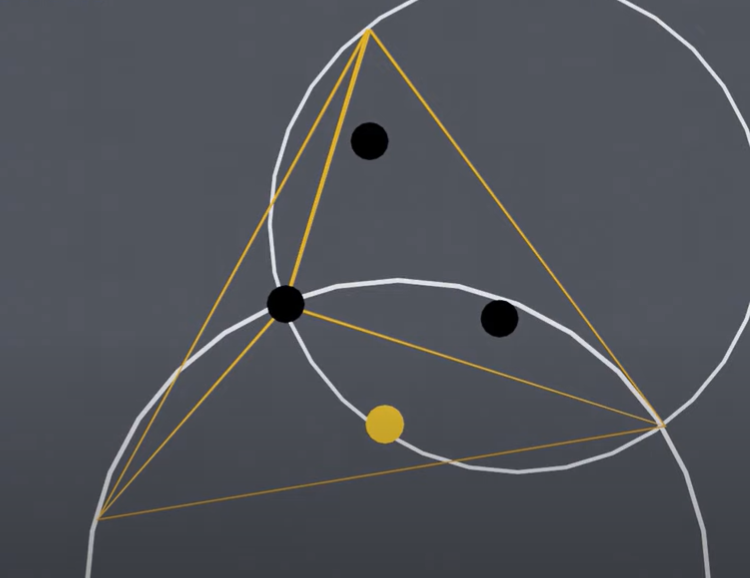
**Randomized Incremental Algorithm**

Randomized Incremental Algorithm will be used to create the Delaunay Triangulation. This algorithm works by incrementally deleting the illegal edges in the triangulation

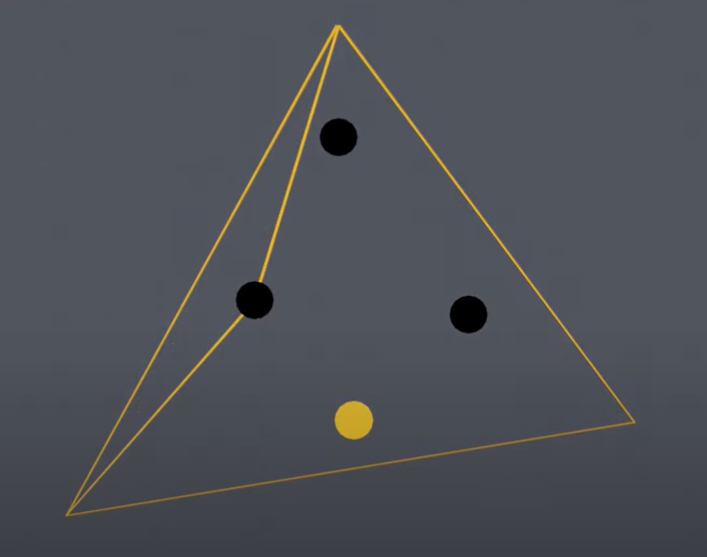
1. Form a super triangle outside the points



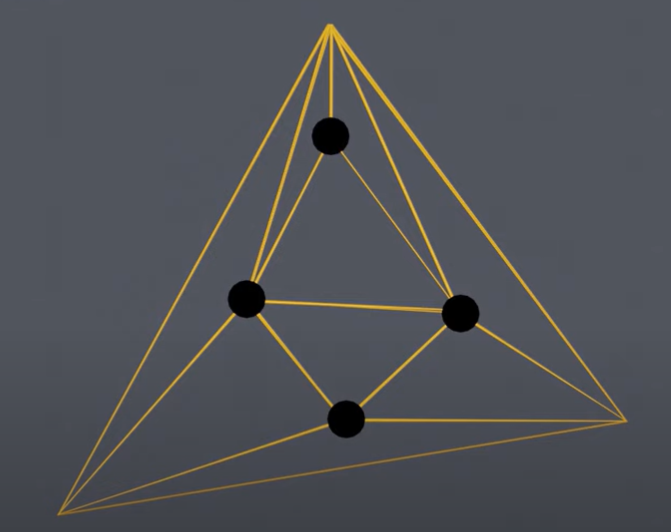
1. Consider triangles whose circumcircles include a point inside “illegal triangles”

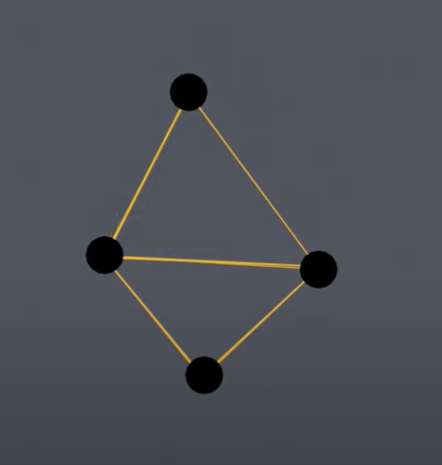


1. Delete edges shared by two illegal triangles



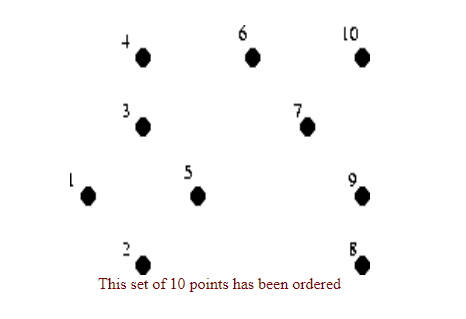
1. Repeat the process for all points
2. Remove the vertices of the super triangle and edges incident upon them



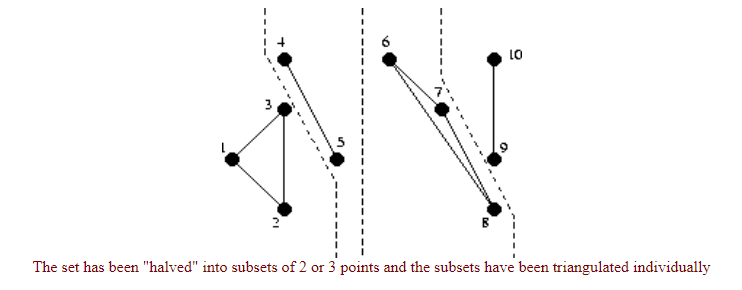


**Divide-and-Conquer Algorithm**

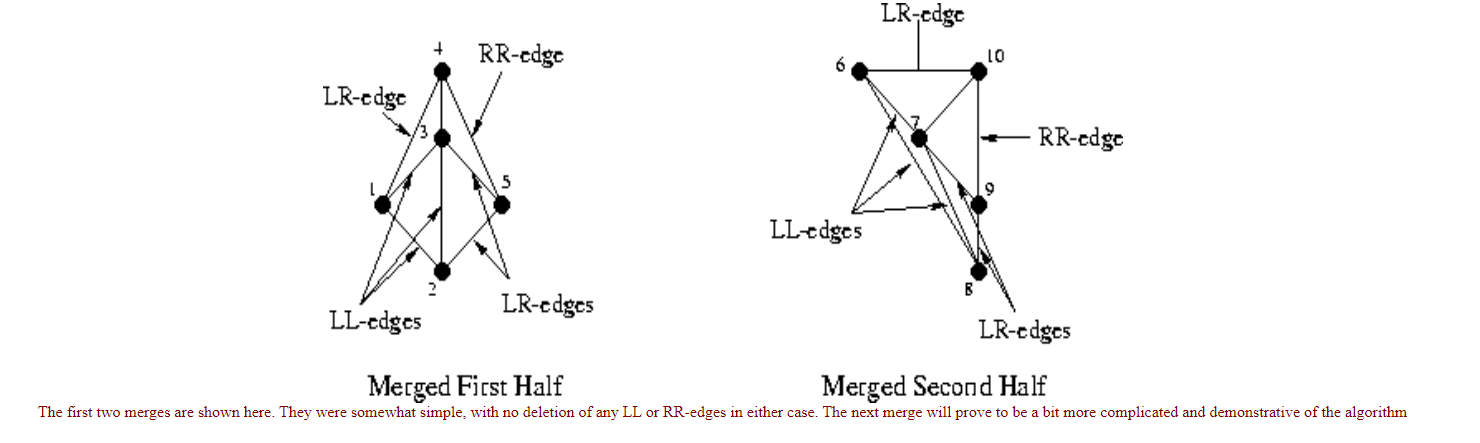
1. Sort the points in increasing order of x-coordinates



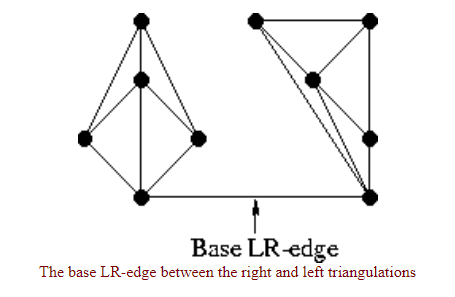
1. Divide the points into subsets of no more than three points



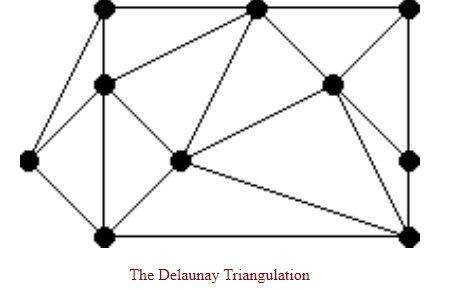
1. Merge the subsets with their former halves



1. When merging, delete the Left-Left and Right-Right edges, and insert Left-Right edges



1. Once the last two halves are merged successfully, the triangulation is complete



**Data Structures**

Storing of points are handled by NumPy library’s data structures

**Implementation Details**

Implementation is done in Python 3. NumPy library is used for storing points and various other tools. Matplotlib library is used for visualizing the triangulations. Tkinter library is used for creating the GUI for the program.

The source code for the Randomized Incremental Algorithm [1] and Conquer and Divide Algorithm [2] are taken from external sources.

Plotting of the triangulations and the GUI are implemented in the main class.

Findings

The triangulations with the Divide and Conquer Algorithm were noticeably faster than the triangulations done with the Randomized Incremental Algorithm.

References

[1] <https://github.com/alexbaryzhikov/triangulation>

[2] <https://github.com/jmespadero/pyDelaunay2D>