MCS 481 Project Three : Arrangements, Delaunay Triangulations, Windowing Queries
due Wednesday 23 April 2025 at 2pm

The goal of this project is to use open source software to explore properties of algorithms we have seen to construct line arrangements, Delaunay triangulations, and to run windowing queries.

0. Software

CGAL is well documented and it is recommended to start working from the examples that come with the installation of the library. For this project you may either use the C++ version, or the Python bindings generated via swig, known as `cgal-swig-bindings`. Delaunay trianguations are available in the spatial module of scipy.

1. Line Arrangements

In `/examples/Arrangement_on_surface_2` we find the program `vertical_ray_shooting.cpp` to answer point queries to solve the discrepancy problem we considered in Chapter 8. Examine whether the implementation can solve the discrepancy problem in time $O(n^2)$ for n points.

In reporting times, distinguish between the time it takes to construct an arrangement and the time a point query takes.

2. Delaunay Triangulations

The program `terrain.cpp` in the directory `/examples/Triangulation_2` in the source code distribution of CGAL illustration the computation of a Delaunay triangulation. The program `voronoi.cpp` builds a Delaunay triangulation from the dual of a Voronoi diagram.

Consider the following question. The complexity of computing a Delaunay triangulation of n points is expected to be $O(n \log(n))$. Take several point configurations in random position and report how long it takes to build a Delaunay triangulation. Make sure to take sufficiently large data sets.

3. Windowing Queries

The program `segment_tree_map_2.cpp` in the folder `RangeSegmentTrees` constructs a segment tree and performs a windows query.

Construct trees for sufficiently large point sets of size n. Report the observed times for the construction of the data structure. Do you observe the $O(n \log(n))$ as n grows?

4. the Deadline is Wednesday 23 April 2025 at 2pm

Upload your answers to gradescope before 2pm on Wednesday 23 April 2025. Your solution consists of

1. The code for the programs you used in the experiments.
2. Tables with properly formatting of the essential characteristics for each experiment: the dimension of input and output, and the running time.
3. Your interpretation of the results and conclusions.

You may work in pairs for this project.

If you work in a pair, then one member of the pair must email the name of the partner to `janv@uic.edu` before 5pm on Wednesday 16 April.

As an alternative to the prescribed Project Three on linear arrangements, Delaunay triangulations, and windowing queries, there is the option to select a topic:

MCS 481 Project Three : a topic in computational geometry
due Wednesday 23 April 2025 at 2pm

The topic in the last project is open to your own interest and could vary between a programming project and a theoretical study. In particular, you could select to do one of the following:

1. Write your own code to explore an algorithm we covered in class.
2. Use CGAL to explore algorithms not covered in the projects.
3. Use other software, e.g.: SageMath, Qhull, for computations.
4. Select a theoretical topic, from a section we did not cover in the textbook.
5. Study a paper from the computational geometry literature.

You must announce to me in an email to `janv@uic.edu` the selection of your topic by Monday 14 April. Write a paragraph or two stating the goals of the project.

4. the deadline is Wednesday 23 April 2025 at 2pm

The report you submit is a final draft if you opt for a final exam instead of a final project presentation.

If you will present a final project in the last week of classes, then submit a first complete draft on Monday 23 April at 2pm, and then the day of the final exam is the deadline for the final report.

This project must be an individual effort. Collaborations are not allowed.
If you have questions or difficulties with the project, feel free to come to my office for help.