Programming Assignment

Deadline: 21 April 2023, 17:00

Question 1 *(10 points)*

Line segment intersection: Implement in C++, the sweep-line algorithm for line segment intersection. You are allowed to use external libraries/implementations of the data structures you need in the algorithm, but you should implement your own segment-segment intersection function. Your code must deal with all degeneracies successfully.

Investigate the behavior of the running time of the algorithm you implemeted, via the following questions:

1. How does the time increase as the number of segments increases? Generate segments with both their end points on a circle chosen uniformly at random. Run the program and report the user cpu time and the number of intersection points. Depending on the speed of your computer, you may have to generate fairly large number of points. Interpret if the timings confirm the $O((n+I)\log(n))$ complexity.
2. Is the algorithm robust? Generate the following configurations of line segments:
   - (a) The endpoints of all segments lie on a circle and every segment passes through the center of the circle, so the center of the circle is an intersection point of high multiplicity, but it is the only intersection point.
   - (b) Take the highly degenerate configuration of line segments from above, and slightly shift one endpoint so each segment gets very close to the center of the circle, but misses it. Verify the correctness of the output for both outputs. For the second shifted configuration, experiment with various small values of the shifting the end points. As with the first assignment, report the number of intersection points as well as the user cpu time for each run.

Question 2 *(10 points)*

Computing the Overlay of two Maps: The creation of a planar subdivision stored as a doubly-connected edge list is illustrated by the program

"/CGAL-4.13/examples/Arrangement on surface 2/overlay.cpp".

While the space complexity to intersect n line segments is $O(n)$, the space complexity for the map overlay problem must be higher than this. Use a CGAL program to generate an example representing the worst-case space complexity. Finally, answer the following:

1. As the size of the input dimension n grows, what is more likely to be the bottleneck for the map overlay problem: time or space? Use your experiments to illustrate your answer.

All your observations must be well-documenetd in your report.
In particular, the report should document the time required by your code for each test dataset.
Provide a short, one-paragraph analysis of your results along with the table showing the times.
Submit your report as a single PDF document along with the source code of your solution (in the Appendix) via E-mail before the deadline.