CSC 437/537 - Geometric Algorithms

Assignment 2

Due: Tuesday, September 21st at 11:59pm

Line Segment Intersection

1) Implement in Python the Bentley-Ottmann algorithm for the line intersection problem. Formulate some test cases to show that the time complexity depends on the output and measure the performance. You can use the code provided to start.
2) Let S be a set of n circles in the plane. Describe a plane sweep algorithm to compute all intersection points between the circles. (Because we deal with circles, not discs, two circles do not intersect if one lies entirely inside the other.) Your algorithm should run in $O((n + k) \log n)$ time, where k is the number of intersection points.

    Some additional notes:

    Your algorithm must be correct even though one circle may be nested within another without intersecting

    Explain clearly (1) what the sweep line status stores and what data structure is used to store this information and (2) what future events are stored and what data structure is used.

    You may assume that you have access to whatever primitive operations that you need in constant time. For example, if you want to determine (a) whether two circles intersect, (b) the coordinates of an intersection, (c) the intersection of a line with a circle, (d) whether a point is contained within a circle's interior, etc., you may simply assume the existence of a function that runs in O(1) time.

Map Overlay

1) Implement in Python a plane-sweep algorithm to compute the overlay of two planar subdivisions represented as DCELs. You can use the DCEL code provided to start.
2) Modify the program to compute the intersection of two planar subdivisions similar to the birch and wolves example in class.

Submit your code, your plots, and any screen captures that demonstrate that your programs work. For any proofs, arguments or pseudocode you can use a word processor or free-hand writing and drawing as long as it is clear and readable.