CMSC 754: Lecture EX1
Review for the Midterm Exam

Midterm Exam: Will be Thu, Mar 12 in class. The exam will be closed-book and closed-notes, but you are allowed one sheet (front and back) of notes.

We have discussed a number of algorithms and data structures that are useful for solving geometric problems. Here is a list of the main topics we have presented so far.

Convex hulls: Convexity and convex hulls algorithms including Graham's scan ($O(n \log n)$), divide-and-conquer ($O(n \log n)$ time), Jarvis march ($O(hn)$ time), and Chan's algorithm ($O(n \log h)$ time). The latter two algorithms are examples of *output sensitive* algorithms, whose running time is a function of output size (h being the number of vertices on the final hull).

Line Segment Intersection: We presented an $O((n + m) \log n)$ time *plane sweep algorithm*, where m is the number of intersection points. This introduced the important algorithm-design concept called *plane-sweep*.

Intersection of Halfplanes: We presented an $O(n \log n)$ time divide-and-conquer algorithm. The key is a merging step that intersects two convex polygons using plane-sweep. (This was not discussed in class, but it appeared in the lecture notes.)

Polygon Triangulation: We presented a simple $O(n \log n)$ time for polygon triangulation. This was based on two pieces: an $O(n)$-time algorithm for triangulating an x-monotone polygon and an $O(n \log n)$ time algorithm to decompose a simple polygon into x-monotone pieces. Both algorithms are based on plane-sweep. (Note that there exists an $O(n)$-time algorithm for simple polygon triangulation, but it is quite complicated.)

Point-line duality: We demonstrated a *dual transformation* that maps a point (a, b) in $\mathbb{R}^2$ to the (nonvertical) line $y = ax - b$ and vice versa. We discussed how numerous affine (point-line) properties are preserved by this transformation. We observed that problems involving lines/points can be converted to an equivalent problem involving points/lines through this transformation. As an example, we showed that computing lower and upper envelopes of a set lines can be reduced to the problem of computing the upper and lower convex hulls of the dual points.

Linear Programming: We presented a *randomized incremental algorithm* for linear programming in spaces of constant dimension d. The algorithm runs in $O(d!n)$ expected time. The randomized incremental method involves inserting halfplanes at random and then updating the optimal point with each new insertion. We introduced the technique of *backwards analysis*, which analyzing the expected time of each step under the assumption that the *last* insertion is random. In the homework assignment, we explored a number of applications of low-dimensional linear programming.

Trapezoidal Maps: We introduced the useful planar decomposition of a collection of line segments called the *trapezoidal map*. We presented a randomized incremental $O(n \log n)$ expected-time algorithm for computing the trapezoidal map of n line segments.

Planar Point Location: By building a *history DAG* for the trapezoidal-map construction process, it is possible to answer point location queries in $O(\log n)$ time and $O(n)$ space. This data structure records the history of the rebucketing decisions made by the construction algorithm.

DCEL (Not covered): The material on the doubly-connected edge list (DCEL) from the last lecture will *not* be covered on the exam.