Computational Geometry - Homework II

K. Subramani
LCSEE,
West Virginia University,
Morgantown, WV
{ksmani@csee.wvu.edu}

1 Instructions

1. The homework is due on March 28, in class. Each question is worth 5 points.

2. Attempt as many problems as you can. You will be given partial credit, as per the policy discussed in class.

2 Problems

1. **Linear Programming:** A simple polygon $\mathcal{P}$ is said to be star-shaped, if it contains a point q, such that for all points $p \in \mathcal{P}$, the line segment $\bar{pq}$ is contained in $\mathcal{P}$. Devise a linear time algorithm to decide if a polygon is star-shaped.

2. **Orthogonal Range Searching:** Let $\mathcal{P}$ denote a set of n axis-parallel rectangles in the plane. Given a query rectangle $[x : x'] \times [y, y']$, the goal is to report all the rectangles in $\mathcal{P}$ that are completely contained in the query rectangle. Describe a data structure for this problem that uses $O(n \cdot \log^3 n)$ storage and answers queries in time $O(\log^4 n + k)$, where k is the number of reported answers.

3. **Point Location:** Give an example of a set of n line segments and the order on them that makes the randomized trapezoidal algorithm create a search structure of size $\Theta(n^2)$ and having query time $\Theta(n)$, in the worst case.

4. **Point Location:** Describe a deterministic algorithm to compute the trapezoidal map of a set of n non-crossing segments in time $O(n \cdot \log n)$. *Hint: Use a variant of plane sweep.*

5. **Voronoi Diagrams:** Let $\mathcal{P}$ denote a set of n points in the plane. Give an $O(n \cdot \log n)$ time algorithm to find for each point p in $\mathcal{P}$, another point in $\mathcal{P}$, that is closest to it.