CMPS 3130/6130 Computational Geometry – Spring 15

4/23/15

Extra Credit Homework

Due **5/5/15** before class

1. **Minkowski sums (5 points)**
   Let A and B be two **convex** sets in $\mathbb{R}^d$. Prove that $A \oplus B$ is convex.

2. **Shortest paths (10 points)**
   (a) (5 points) Consider a set of disjoint polygonal obstacles in $\mathbb{R}^2$ with n edges total. Show that for any $a, b \in \mathbb{R}^2$ the number of segments on an obstacle-avoiding shortest path from a to b is bounded by $O(n)$.

   (b) (5 points) Give an example in the plane with n line-segment obstacles and two points a, b such that the number of different shortest paths between a and b is exponential in n. *(Hint: It might help to assume that the line segments are open, i.e., endpoints are no obstacles. This means that two line segments can touch in a point, and a path is allowed to pass through this point.)*

3. **Nesting segment trees and range trees (10 points)**
   In class we used a *segment-range* tree to solve the 2-dimensional windowing problem. This two-level tree consists of a segment tree as the primary tree, and it stores in each node of the primary tree a link to a secondary tree which is implemented as a range tree.

   Now consider defining a *range-segment* tree which has a range tree as the primary tree and segment trees as the secondary trees. We can also define a *segment-segment* tree in a similar way, and *range-range* trees we have already studied in class.

   Compare all four data structures and argue what kinds of problems each can be used to solve. Analyze and compare the query times, construction times, and space complexities.

4. **KD-trees (15 points)**
   (a) (6 points) Describe an algorithm to construct a d-dimensional kd-tree for a set P of n points in $\mathbb{R}^d$. Prove that the algorithm takes $O(n \log n)$ time and that the tree can be stored in $O(n)$ space. Assume d is constant.

   (b) (2 points) In the proof for the query time for a 2-dimensional range tree we used the recurrence $Q(1) = 1$ and $Q(n) = 2 + 2Q(n/4)$ for $n \geq 2$. Prove that this recurrence solves to $Q(n) = O(\sqrt{n})$.

   (c) (3 points) Describe a query algorithm for performing an orthogonal range query in a d-dimensional range tree.

   (d) (2 points) For $d = 3$, show that your query algorithm runs in time $O(n^{\frac{2}{3}} + k)$. For this, develop a recurrence for $Q(n)$ and solve it.

   (e) (2 points) Now generalize your query time analysis for general d to show that your query algorithm runs in time $O(n^{\frac{d-1}{d}} + k)$. Assume d is constant.