4/13/17

8. Homework

Due **Thursday 4/20/17** before class.

1. **Interval Counting (6 points)**
   Develop a simple data structure that stores a set I of n intervals on the line such that for a given query point q the number of intervals containing q can be *counted* in $O(\log n)$ time. Your data structure should use $O(n)$ storage. Analyze storage, preprocessing time, and query time.

2. **Nesting 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 each node of the primary tree stores a link to a secondary 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.

3. **KD-trees (14 points)**
   (a) (5 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 kd-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.