COS 423: Homework Assignment 1

due Wed 02/29/2012

Problem 1

Suppose you are given a set S of n segments parallel to the X-axis: $s_1, \ldots, s_n$. Each s_i joins two points of the form (x_i, y_i) and (x'_i, y_i); all x_i, x'_i, y_i are distinct. A segment tree will allow you to store these segments so as to support the following *vertical query*: Given a vertical segment $q : (x, y), (x, z)$, count how many segments s_i intersect q. The query time is $O(\log^2 n)$ and the storage is $O(n \log n)$.

1. Modify the algorithm so that it supports insertions and deletions of segments, all taken from the set S, with a query/insert/delete time of $O(\log^2 n)$. You may assume you know the full set S ahead of time.

2. Modify the algorithm so that the amortized query/insert/delete time is still $O(\log^2 n)$ but the storage $O(n + m \log n)$, where $m \leq n$ is the number of segments currently in the set at the time of the query. [Hint: once in a while rebuild the whole structure from scratch.]

3. How would you store n axis-parallel rectangles, using a segment tree, so that given a query point (x, y), you can tell in $O(\log n)$ time whether it lies inside the union of the rectangles or outside?

Problem 2

Consider a graph consisting of a single path $v_1, \ldots, v_n$, each v_i with its own catalog C_i. Design a simple fractional cascading structure based on two passes and give a tight analysis. Treat the gap size as a parameter.

Problem 3

Design a fractional cascading structure where the cascading travels around the same cycle several times. How many times around the cycle can you achieve as a function of the number of input keys and the size of the cycle? Feel free to experiment on a computer.

Problem 4

Given an n-by-n matrix M with n^2 distinct real entries, and each row (and column) appearing in increasing order, find an $O(n)$-time algorithm to check whether a given number x is in it, ie, whether there exists i, j such that $x = M_{ij}$. No hashing allowed.