CS 266 Homework 5

Zachary DeStefano, PhD Student, 15247592

Due Date: May 15

Problem 10.1

In Section 10.1 we solved the problem of finding all horizontal line segments in a set that intersect a vertical segment. For this we used an interval tree with priority search trees as associated structures. There is also an alternative approach. We can use a 1-dimensional range tree on the y-coordinate of the segments to determine those segments whose y-coordinate lies in the y-range of the query segment. The resulting segments cannot lie above or below the query segment, but they may lie completely to the left or to the right of it. We get those segments in O(logn) canonical subsets. For each of these subsets we use as an associated structure an interval tree on the x-coordinates to find the segments that actually intersect the query segment. q a. Give the algorithm in pseudocode. b. Prove that the data structure correctly solves the queries. c. What are the bounds for preprocessing time, storage, and query time of this structure? Prove your answers.

Problem 10.6c

Let I be a set of intervals on the real line. We want to be able to count the number of intervals containing a query point in O(logn) time. Thus, the query time must be independent of the number of segments containing the query point.

Describe a data structure for this problem based on a simple binary search tree. Your structure should have O(n) storage and O(logn) query time. (Hence, segment trees are actually not needed to solve this problem efficiently.)

Problem 14.6

In this chapter we called a quadtree balanced if two adjacent squares of the quadtree subdivision differ by no more than a factor two in size. To save a constant factor in the number of extra nodes needed to balance a quadtree, we could weaken the balance condition by allowing adjacent squares to differ by a factor of four in size. Can you still complete such a weakly balanced quadtree subdivision to a conforming mesh such that all angles are between 45 and 90 by using only O(1) triangles per square?

Problem 14.12

Suppose we have quadtrees on pixel images I1 and I2 (see the previous exercise). Both images have size 2k 2k, and contain only two intensities, 0 and 1. Give algorithms for Boolean operations on these images, that is, give algorithms to compute a quadtree for I1 I2 and I1 I2. (Here I1 I2 is the 2k 2k image where pixel (i, j) has intensity 1 if and only if (i, j) has intensity 1 in image I1 or in image I2. The image I1 I2 is defined similarly.)