

COMPUTATIONAL GEOMETRY

Homework Set # 7

Due in class on Monday, May 9, 2011. *Reminder: Show your reasoning!* Quiz 4 will be 5:20-5:40 on May 11, covering hw6 and hw7.

Recommended Reading: BCKO/BKOS: Chapter 10.

DO ANY 3 OF THE FOLLOWING 5 PROBLEMS.

(1). We want to solve the following query problem: Given a set S of n disjoint line segments in the plane, determine those segments that intersect a vertical ray running from a point $q = (q_x, q_y)$ vertically upwards to infinity. Describe a data structure for this problem that uses $O(n \log n)$ storage and has a query time of $O(\log n + k)$, where k is the number of reported answers. Explain how your method achieves the claimed bounds.

(2). Build the segment tree for the following intervals: $[0 : 6]$, $[1 : 5]$, $[2 : 9]$, $[4 : 9]$, $[2 : 12]$, $[5 : 11]$, $[7 : 11]$, $[6 : 11]$, $[10 : 11]$. Draw your solution in the same manner as done in Figure 10.9 of the text, with arrows showing the canonical sets as in the figure.

(3). Let S be a given set of n axis-parallel rectangles in general position in the plane. Our goal is to compute very efficiently the *depth*, $\max_{p \in \mathbb{R}^2} \text{depth}(p)$, of the set S , where $\text{depth}(p)$ is the number of rectangles of S that contain p . (We are computing the overall “depth” of the set S , not just the depth at any one point p .) Describe in some detail how a sweep algorithm can compute the depth efficiently. (Hint: a segment tree may be useful.) What is the running time and what is the space used (in big-Oh)?

(4). *Problem 10.9, BCKO/BKOS.* Let I be a set of intervals on the real line. We want to store these intervals such that we can efficiently determine those intervals that are completely contained in a given interval $[x : x']$. Describe a data structure that uses $O(n \log n)$ storage and solves such queries in $O(\log n + k)$ time, where k is the number of answers. *Hint:* Use a range tree.

(5). *Problem 10.10, BCKO/BKOS.* Again we have a collection I of intervals on the real line, but this time we want to efficiently determine those intervals that contain a given interval $[x : x']$. Describe a data structure that uses $O(n)$ storage and solves such queries in $O(\log n + k)$ time, where k is the number of answers. *Hint:* Use a priority search tree.