

### Assignment 6

Due: Thursday, April 1, 2010 (before class)

**Note:** For both problems start off by describing the overall approach used followed by a high level description of the algorithm. Solutions consisting of only code or code-like descriptions will not get graded.

1) (25 pts.) You are given  $n$  line segments  $s_1, s_2, \dots, s_n$  in 2-D. Each line segment  $s_i$  is either a horizontal or vertical line segment and has a color, either red or blue. A line segment is defined by its two endpoints; i.e.,  $s_i = (p_i, q_i) = ((x_{i,1}, y_{i,1}), (x_{i,2}, y_{i,2}))$  with either  $x_{i,1} = x_{i,2}$  or  $y_{i,1} = y_{i,2}$ , and its color  $c_i$ ,  $1 \leq i \leq n$ . No two line segments have identical endpoints and none corresponds to a point. Consider the problem of determining whether every vertical line segments intersects with at least one blue and at least one red horizontal line segment.

(i) Show that this problem has an  $\Omega(n \log n)$  lower bound.

Hint: Make a reduction using one of the following problems known to have an  $\Omega(n \log n)$  lower bound: sorting, element uniqueness, set equality, set inclusion. The lower bound holds when the quantities are integers (for algorithms that make no assumption about the size of the integers).

(ii) Describe and analyze an algorithm having a time complexity matching the lower bound. Make sure to explain any of the data structures used. You may find it useful to read Section 33.2, but the solution you present needs to take advantage of the fact that the line segments are horizontal or vertical.

2) (25 pts.) You are given an  $n$ -vertex free tree  $T$  represented as a rooted tree with root  $r$  (e.g., stored in array  $\Pi$ ). You are to place resources at vertices such that (i) every vertex is at most distance 2 from a resource and (ii) the total number of resources used is a minimum. Describe an efficient algorithm for placing resources. Analyze its time complexity and address its correctness.

Hint: Consider first the problem of placing a minimum number of resources so that every vertex is at most distance 1 from a resource.