CS381 **Homework 5** **Fall 2012**

Question 1. (20 points) This question is about the algorithm we covered in class for a longest increasing subsequence (if you missed that lecture or did not take notes, you can find much online material about this topic, for example see http://en.wikipedia.org/wiki/Longest_increasing_subsequence). Run the algorithm on the following input and state which longest increasing subsequence it computes:

$$15, 7, 5, 10, 8, 4, 13, 11, 2$$

Note that there are many increasing subsequences of same length: We are interested in only one of them, the one that is computed by the algorithm we gave in class.

Question 2. (20 points) Let $\mathcal{A}$ be a comparison-based algorithm that correctly computes the median of a set of $2n+1$ distinct elements. The input to $\mathcal{A}$ could correspond to any of the $(2n+1)!$ possible permutations of $2n+1$ numbers, and the only way for $\mathcal{A}$ to gain information about the input is by making comparisons. For each of the following two statements, state whether that statement is true or false. Then briefly justify your answers.

1. For every one of the possible inputs, $\mathcal{A}$ must use at least $2n$ comparisons.
2. There exists an input for which $\mathcal{A}$ must use at least $2n$ comparisons, but $\mathcal{A}$ might use fewer comparisons for some other inputs.

Question 3. (20 points) Prove an $\Omega(n \log n)$ time lower bound for the following problem. The input consists of a set S of $2n+1$ distinct numbers, $S = a_1, \ldots, a_{2n+1}$. The output is "yes" if and only if the median (call it w) of S is such that for every δ, if $w + \delta \in S$ then $w - \delta \in S$ (in other words w is a "center of symmetry" for S).

Hint. Use a reduction from the set-equality problem, and keep in mind that, in an instance of set equality, the two sets can contain both positive and negative numbers.

Question 4. (20 points) Let D be a data structure that performs in time $T(n)$ each Insert(x) operation and each ExtractMin operation, without any restrictions on x (i.e., without requiring that x be a small integer, or any similar constraint that might make the problem easier). Recall that an Insert(x) has the effect of adding element x to the data structure, an ExtractMin operation simultaneously returns and deletes the smallest element in the data structure. For example, suppose the following operations are performed on an initially empty data structure:

Insert(5.5), Insert(2.1), ExtractMin, Insert(19.7), ExtractMin.

Then the output of the first ExtractMin is 2.1, that of the second ExtractMin is 5.5, and at the end of the above operations the data structure contains only the element 19.7.
Prove an $\Omega(\log n)$ lower bound for $T(n)$.

Question 5. (20 points) The first part of this question is about the $O(n \log n)$ time intersection-detection algorithm we gave in class, when the input to the algorithm consists of the line segments

$S1, \ldots, S7$ that are shown on the next page. Recall that the algorithm stops as soon as it finds an intersection, and reports that intersection. Which intersection is reported by the algorithm ?

The second part of this question is about the "enumerate-all-intersections" version of the algorithm (recall that this was a modified version of the detection algorithm). List the intersections in the order produced by the intersection-enumeration algorithm.

Note. The textbook contains only the intersection-detection version of the algorithm, so if you missed the lecture or did not take notes you can find the enumeration version of the algorithm at the online lecture notes at http://www.cs.wustl.edu/ pless/506/l4.html or by reading the journal paper using the IEEE online library that is accessible from purdue.edu: J. L. Bentley and T. Ottmann., Algorithms for reporting and counting geometric intersections, IEEE Transactions on Computers C28 (1979), 643–647.

Date due: October 18, 2012

Starting Oct 11, the class will meet in LWSN B155 rather than GRIS 180.

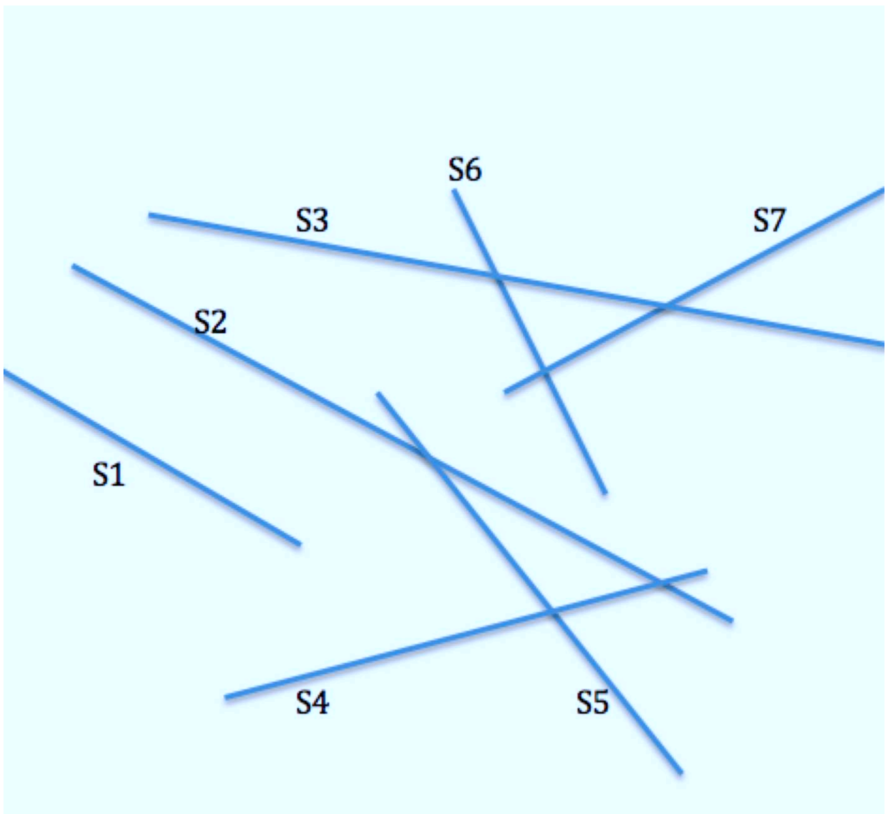

Figure 1: The input segments for question 5