Junior-Prof. Dr. Robert Elsässer, Marco Muñiz, Phillip Heidegger WS 2009/2010

Algorithms Theory, Assignment 1

`http://lak.informatik.uni-freiburg.de/lak_teaching/ws09_10/algo0910.php`

The solutions can be submitted in English or German. hand in by **5. Nov. 2009, 4 p.m.**

Exercise 1.1 - Minimum and maximum [Points: 1+2+3]

Let $\mathcal{U}$ be a universe with a total order $\prec$. Define $S = \{e_1, \ldots, e_n\}$ with $n = 2^i$, where $i \in \mathbb{N}_{\geq 1}$ and $e_1, \ldots e_n$ are some elements from $\mathcal{U}$.

1. Describe an iterative procedure (in pseudocode) that computes the maximum and minimum in S. How many comparisons ($\prec$) are used? Prove your answer.
2. Design an algorithm using the divide-and-conquer approach that only needs $\frac{3}{2}n - 2$ comparisons. Provide the corresponding pseudocode and analyze its runtime complexity.
3. Now assume that our set consists of natural numbers and the comparison operation is $\leq$. Analyze the previous algorithms and compute all comparisons that occur during the execution of the program (not only the comparisions between the elemenst of S).

Exercise 1.2 - Closest Pair [Points: 4+3]

1. Let the set $S = \bigcup_{i \in [1 \ldots 8]} \{p_i\}$ contain the following points

$p_1 = (25, 6)$	$p_2 = (12, 10)$	$p_3 = (0, 3)$	$p_4 = (7, 12)$
$p_5 = (8, 2)$	$p_6 = (13, 3)$	$p_7 = (9, 11)$	$p_8 = (1, 0)$

   Find the *closest pair* using the strategy presented in the lecture and state all intermediate results.

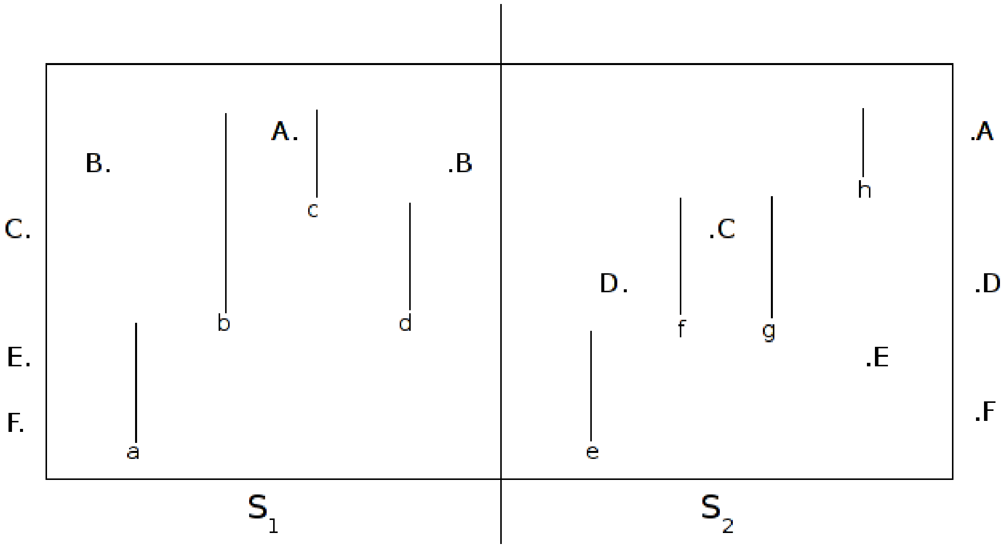
2. Some algorithms assume that for all $p = (p_x, p_y)$ there is no other point $q = (q_x, q_y)$ such that $p_x = q_x$ or $p_y = q_y$.

   What kind of problems can arise if this condition is not fulfilled?

Exercise 1.3 - Geometric Divide-and-Conquer [Points: 4]

Let $a, \ldots, h$ be vertical line segments and $A, \ldots, F$ be horizontal line segments as given in the figure below. Each horizontal segment is defined by its left and right endpoint (denoted by $X.$ and $.X$, respectively, where $X \in \{A, \ldots, F\}$). Furthermore, let S be the set of all vertical line segments together with all (left and right) endpoints of the horizontal line segments. Assume that all pairs of intersecting line segments should be determined using the divide-and-conquer procedure presented in the lecture. The divide step yields the partitioning below, where $S = S_1 \cup S_2$. Specify the output of the two calls ReportCuts(S_1) and ReportCuts(S_2), the sets $L(S_1), L(S_2), R(S_1), R(S_2), V(S_1), V(S_2), L(S)$ and $R(S)$, as well as the intersections identified in the merge step. Then, determine the output of ReportCuts(S).

Notation: $L(S) = \{P \mid P. \in S \wedge .P \notin S\}$, $R(S) = \{P \mid P. \notin S \wedge .P \in S\}$ and $V(S) = \{p \mid p \in S\}$. The intersection of a horizontal line segment P with a vertical line segment p is denoted by the ordered pair (P, p). Therefore, the set ReportCuts(S) consists of some ordered pairs (P, p)



Exercise 1.4 - Closest Pair - Memory [Points: 3]

State the space requirement of the algorithm NächstesPaarRec presented in the lecture.
Use the Θ notation to specify how much space is needed by the algorithm and prove your result by induction.

Hint: You may assume that $n = 2^i$ for some $i \in \mathbb{N}_{\geq 1}$, where n is the total number of points.