Note: You should solve all the problems in the Mandatory Part, and one of the two problems in the Optional Part.

**Mandatory Part**

**Problem 1. Intersection check** Given a set $S$ of $n$ vertical segments $s_1, \ldots, s_n$ in the plane, construct a data structure that uses $O(n \log n)$ space, that given a horizontal segment $s$, checks if $s$ intersects any of segments in $S$ in $(\log O(1)n)$ time.

**Problem 2. Textbook, exercise 8.15, p. 182** Let $S$ be a set of $n$ segments in the plane. A line $l$ that intersects all segments in $S$ is called a stabber for $S$. Give an $O(n^2)$-time algorithm to decide if there exists a stabber for $S$.

**Problem 3. Textbook, exercise 9.11, p. 209** A Euclidean Minimum Spanning Tree (EMST) of a set $P$ of $n$ points in the plane is a tree over the vertex set $P$ of a minimum total edge length. Show that the set of edges of a Delaunay triangulation of $P$ contains a EMST of $P$. Does this imply an $O(n \log n)$-time algorithm for computing EMST?

**Optional Part**

**Theoretical Problem. Textbook, exercise 8.4, p. 181** Let $L$ be a set of $n$ lines in the plane. Give a $O(n \log n)$-time algorithm to compute an axis-parallel rectangle that contains all the vertices of the arrangement of $L$.

**Programming Problem.** Implement a Java Applet that solves the motion planning problem, for the special case of rectangles.

More specifically, the scene consists of a set of non-intersecting rectangles. The robot is also a rectangle, and has to go from an initial point $A$, to some final point $B$. The solution does not have to be the shortest path, and the algorithm does not have to be optimal.

The applet should also provide a way for the user to specify the input. Graphical specification is strongly preferred. It should demonstrate the algorithm, in addition to solving the problem. Feel free to add any extra bells and whistles.