This exercise sheet deals with scanline algorithms for solving geometrical problems. Every description of a scanline algorithm should address the following aspects.

a) *Stopping points.* In which direction is the scanline moving? What are the stopping points?

b) *Scanline data structure.* Which objects have to be stored in the data structure? Which operations have to be supported? What is an appropriate data structure?

c) *Update.* What happens if the scanline encounters a new stopping point?

d) *Extracting the solution.* How can the solution be extracted?

**Exercise 12.1  Segment trees.**

Consider a general segment tree for integer intervals with boundaries from \{1, ..., n+1\}. Describe in your own words why no interval needs to be inserted in \(\Theta(n)\) nodes. Give an interval which is inserted in a maximum number of nodes and justify why this number of nodes is maximum. 

*Note:* You may assume that \(n\) is a power of two.

**Exercise 12.2  Piercing orthogonal rectangles.**

You are given \(n\) orthogonal rectangles which may overlap. We are searching a point that maximizes the number of pierced rectangles. A rectangle is pierced by a point if and only if it is lying inside or on the boundary. In the following example, \(P\) is such a point because it pierces four rectangles while no other point exists that pierces more.

a) Describe an efficient scanline algorithm to find a point that maximizes the number of pierced rectangles.

b) What is the running time of your algorithm?
Exercise 12.3  Overlapping triangles.

You are given a set of triangles in the plane. Each triangle is provided by the coordinates of its nodes. For simplicity, we assume that no nodes share the same $x$-coordinate. We want to determine whether the set contains two triangles that overlap.

a) We first determine whether there are triangles who's boundaries intersect (left figure). Describe a scanline algorithm that solves this problem as efficiently as possible. You may assume that no two triangle boundaries touch without crossing. Provide the running time of your algorithm.

b) If there are no triangle boundaries that intersect, we still need to determine whether there is a triangle that is contained in some other triangle (right figure). Describe a scanline algorithm that solves this problem as efficiently as possible. You may assume that no two triangle boundaries touch or intersect. Provide the running time of your algorithm.