

Homework Assignment #2

Computational Geometry (Winter Semester 2020)

Exercise 1

Set R be a set of n axis-parallel rectangles in the plane. Two rectangles *intersect* if there is a point that is contained in both rectangles (including the boundary).

a) A special case of an intersection is an *overlap*, that is, an intersection in which no rectangle completely contains the other one. For the sake of simplicity, we first consider overlaps as in figure 1, that is, the top-right corner of the lower rectangle lies to the right of the top-right corner of the upper rectangle.

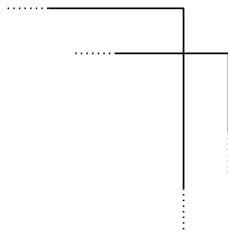


FIGURE 1: A special type of overlap.

Give an algorithm that decides in $O(n \log n)$ time whether R contains two rectangles that overlap in the manner outlined. [5 points]

b) Give an algorithm that decides in $O(n \log n)$ time whether R contains two intersecting rectangles. [5 points]

Hint: It is only important that the algorithm decides whether there is *any* intersection. You do not have to determine the type of the intersection.

Exercise 2

Let P be a finite set of points in the plane. The *largest top-right region* of a point $p \in P$ is the union of all open axis-parallel squares that touch p with their bottom-left corner and contain no point of P .

a) Prove that the largest top-right region of a point is either a square or the intersection of two open half-planes. [2 points]

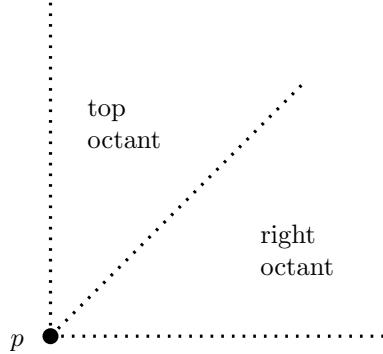


FIGURE 2: Top and right octant of the point p .

b) For every point $p \in P$ and each of the two corresponding octants (see Figure 2), consider which points of P restrict the largest top-right region of p the most. Is the knowledge of these points in both octants enough to determine the largest top-right region of p ? [2 points]

c) Given a set P of n points, compute the largest top-right region of every point in P with total running time $O(n \log n)$. [6 points]

Hint: „Sweep“ the plane twice to determine the points in subexercise b).