Prof. G. Zachmann
J. Boeckers

University of Bremen
School of Computer Science
CGVR Group
May 30, 2025

Summer Term 2025

Assignment on Computational Geometry - Sheet 3

Exercise 1

Give an example of a set S of n non-intersecting line segments in the plane for which a BSP tree of size n exists, whereas any auto-partition of S has size at least $\frac{4n}{3}$.

Exercise 2

Consider this deterministic algorithm for constructing a BSP tree for a set of n non-crossing line segments in the plane. Suppose you are deciding how to slice a cell C of the BSP subdivision. (When we start, C is the entire plane.) If there is a free split*, take it. Otherwise, make a list of the segment endpoints (not fragment endpoints, just endpoints of the original segments) in C, and draw a vertical splitting line through the endpoint with median x-coordinate (breaking ties arbitrarily).

Prove that this algorithm always constructs a BSP tree with at most $O(n \log n)$ fragments.

**free split*: Splitting along a fragment which crosses a region completely i.e. both ends of the fragment lie on already defined splitting lines. It is called free split because its guaranteed to not create more fragments.

Exercise 3

Let's consider the problem of determining whether a query point lies in a complicated polygon. Let's assume it is a simple polygon, that is, its boundary consists of a single cycle of edges, with no self-intersections. We assume the polygon is given as a list of n vertices, in counter-clockwise order on the boundary. One possible data structure we could use to answer point-in-polygon queries is a BSP tree. Argue that it might still require $O(n)$ time, in the worst case, to determine whether a point is in the polygon, if the BSP tree is an auto-partition.