Chair of Computer Science I
Algorithms and Complexity
University of Würzburg

Würzburg, November 25, 2020

Dr. Philipp Kindermann
Dr. Myroslav Kryven

Homework Assignment #5
Computational Geometry (Winter Semester 2020)

Exercise 1

One can use the data structures described in the lecture to determine whether a particular point $(a, b) \in \mathbb{R}^2$ is contained in a given set of points $P \subseteq \mathbb{R}^2$ by performing a range query with range $[a, a] \times [b, b]$. Let $n = |P|$ be the number of given points.

a) Prove that performing such a range query on a kd-tree takes $O(\log n)$ time, using the algorithm from the lecture. **[4 points]**

b) Is it possible to achieve a running time faster than $O(\log^2 n)$, by using such a query on range trees (without fractional cascading, which will be presented in the next lecture)? Prove your answer. **[4 points]**

Exercise 2

In many applications, one wants to do range queries among objects other than points.

a) Let S be a set of n axis-aligned rectangles in the plane. We want to be able to report all rectangles in S that are completely contained in a query rectangle $[x, x'] \times [y, y']$. Describe a data structure for this problem that uses $O(n \log^3 n)$ storage and has $O(\log^4 n + k)$ query time, where k is the number of reported rectangles. **[6 points]**

b) Let P consist of a set of n polygons in the plane. We want to be able to report all polygons in P that are completely contained in a query rectangle $[x, x'] \times [y, y']$. Describe a data structure for this problem that uses $O(n \log^3 n)$ storage and has $O(\log^4 n + k)$ query time, where k is the number of reported polygons. **[6 points]**

Exercise 3

Given a set P of n points and a number $\varepsilon > 0$, we want to compute all pairs of points whose L_∞-distance is at most ε, that is, we want to find each pair p, q in P whose x-coordinates differ by at most ε and whose y-coordinates differ by at most ε.

Design a sweep-line algorithm to solve this problem in $O((k + n) \log n)$ time, where k is the size of the output. Can you solve the problem even in $O(k + n \log^2 n)$ time?

[6 extrapoints]

This assignment is due on December 9th. The solutions will be made available on December 11th.