

Summer Term 2023

## Assignment on Computational Geometry - Sheet 1

Due Date 01. 05. 2023

Please send a digital pdf version of your solution to [weller@informatik.uni-bremen.de](mailto:weller@informatik.uni-bremen.de) until  
**01. 05. 2023**

### Exercise 1 (Balanced Quad Trees, 6 Credits)

Suppose we make the balancing condition for quadtrees more severe: we no longer allow adjacent squares to differ by a factor two in size, but we require them to have exactly the same size. Is the number of nodes in the new balanced version still linear in the number of nodes of the original quadtree? If not, can you say anything about this number?

### Exercise 2 (Point Location in a Quadtree, 6 Credits)

Given any quadtree  $Q$  over the unit square  $[0, 1]^2$ . Given a query point  $q \in [0, 1]^2$ .

Describe an algorithm that finds the deepest node in  $Q$  that contains  $q$ . What is its time complexity?

### Exercise 3 ((Range Queries with Quadtrees,, 8 Credits)

Quadtrees can be used to perform range queries <sup>1</sup>. Describe an algorithm for querying a quadtree on a set  $P$  of points with a query region  $R$ . Analyze the worst-case query time for the case where  $R$  is a rectangle, and for the case where  $R$  is a half-plane bounded by a vertical line.

---

<sup>1</sup> Generally, we use range queries for a set  $S$  of objects to determine which objects from  $S$  intersect with a query object  $R$ , called a range. For example, if  $S$  is a set of points corresponding to the coordinates of several cities, a geometric variant of the problem is to find cities within a certain latitude and longitude range. The range searching problem and the data structures that solve it are a fundamental topic of computational geometry. In orthogonal range searching, the set  $S$  consists of  $n$  points in  $d$  dimensions (in our case here,  $d = 2$ ), and the query consists of intervals in each of those dimensions. Thus, the query consists of a multi-dimensional axis-aligned rectangle.