MA-INF 1203 Discrete and Computational Geometry

Wintersemester 2019/20
Assignment 10

Deadline: **7** January before noon (To be discussed: **7/8**. January 2020)

1 Compressed quadtrees

Consider the points $p_1 = (0.05, 0.01)$, $p_2 = (0.07, 0.01)$, $p_3 = (0.12, 0.15)$, $p_4 = (0.3, 0.3)$, $p_5 = (0.63, 0.68)$, $p_6 = (0.68, 0.68)$ as depicted in Figure 1. Draw the compressed quadtree which stores $p_1, \ldots, p_6$ at its leaves. How many internal nodes including the root node are there? For each internal node v_i indicate the cube $\square(v_i)$ associated with it.

2 Approximate range counting

For any $p \in \mathbb{R}^2$, $r > 0$, let $B(p, r)$ be the Euclidean ball of radius r, centered at p. Consider a set P of n points in $\mathbb{R}^2$, stored in a compressed quadtree. Design an algorithm which, given a query point $q \in \mathbb{R}^2$, radius $r > 0$, and approximation parameter $\epsilon > 0$, returns an integer m such that

$$|B(q, r) \cap P| \leq m \leq |B(q, (1 + \epsilon)r) \cap P|.$$

You can preprocess the compressed quadtree which stores P. The query algorithm must be adaptive to ϵ, meaning that larger values of ϵ should lead to faster running time. Analyze the running time of your algorithm.

3 Construction of the quadtree

a) Consider the recursive algorithm for building a compressed quadtree on n points in $\mathbb{R}^2$: given a non-empty canonical cube, the algorithm subdivides it into at most 4 non-empty canonical cubes, and it recurses on each one of them. Show that this algorithm needs $\Omega(n^2)$ time in general.

b) Design an algorithm for constructing a donut tree (compressed quadtree with holes) and analyze its running time. You may assume that, given a set P of n points, you can compute the following in $O(n)$ time:

   - the smallest canonical square that contains P
   - the smallest canonical square that contains at least $\frac{2^d}{2^d+1} n$ points of P.

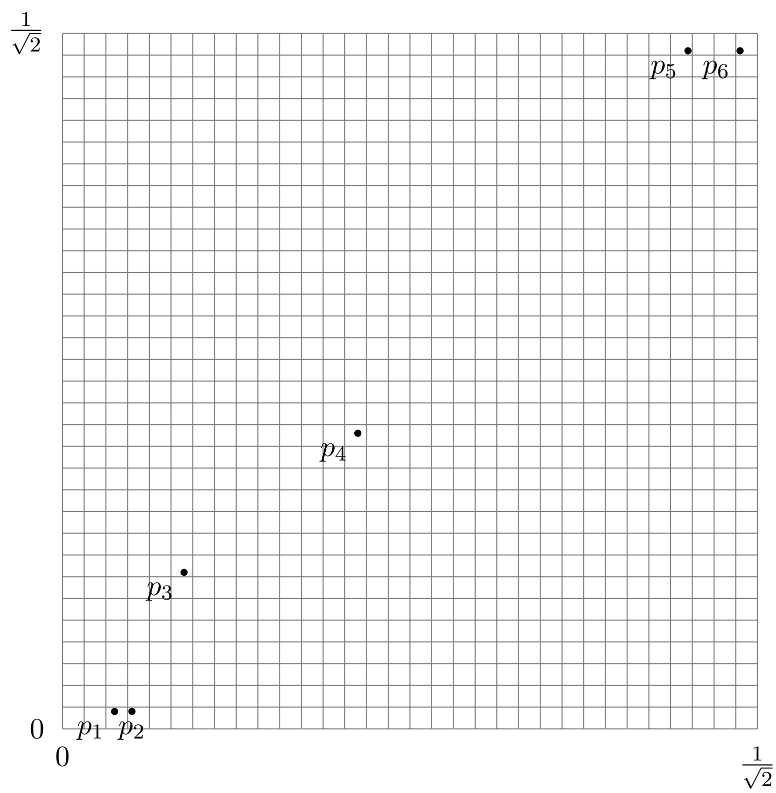

Figure 1