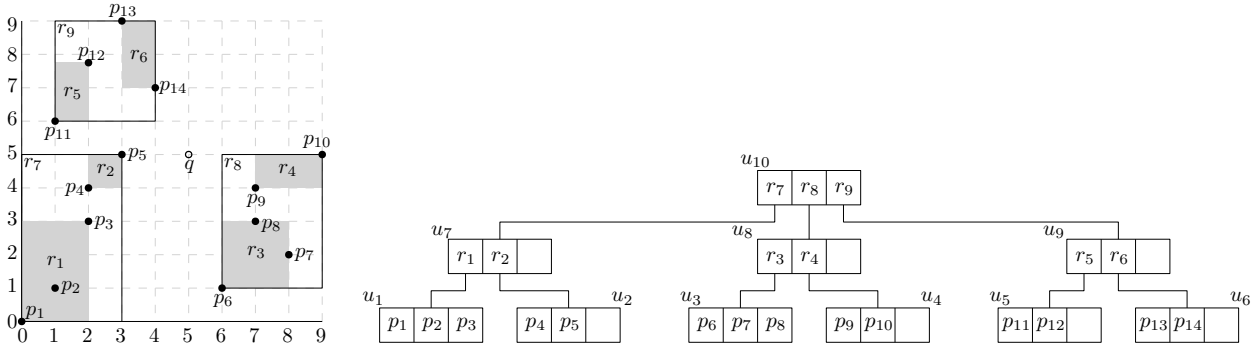
INFS 4205/7205: Exercise Set 3

Prepared by Yufei Tao and Junhao Gan

Problem 1. Consider the dataset $P = \{p_1, \ldots, p_{14}\}$, an R-tree on P, and a nearest neighbor query point q as shown below. Indicate the nodes accessed by the BaB algorithm.



Problem 2. Indicate the nodes accessed by the BF algorithm in the example of the previous problem.

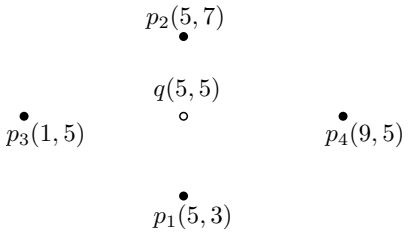
Problem 3. Given an axis-parallel rectangle r and a point q in d-dimensional space, describe an algorithm to compute $mindist(q, r)$ in $O(d)$ time.

Problem 4. Consider a 2D data space where each dimension has range $[0, 1]$. Fix an axis-parallel rectangle $r = [0.5, 0.7] \times [0.5, 0.8]$. Let C be a circle with radius 0.1. Randomly place C such that the center q of C is uniformly distributed in the data space. What is the probability that C intersects r (in other words, $mindist(q, r) \leq 0.1$)?

Problem 5 [k Nearest Neighbor Search]. Let P be a set of d-dimensional points in $\mathbb{R}^d$. Given an integer k and a query point q, a k *nearest neighbor* (NN) *query* returns the k points in P with the smallest (Euclidean) distance to q. This informal definition, however, is ambiguous when some points are equi-distance to q. Formally, a kNN query returns the minimum subset Q of P that satisfies the following two conditions:

- $|Q| \geq k$
- For every point $p \in Q$ and every point $p' \in P \setminus Q$, it holds that $\|p, q\| < \|p', q\|$ (recall that $\|.,.\|$ returns the distance of two points).

For example, in below figure, for $k = 1$, $Q = \{p_1, p_2\}$, and for $k = 3$, $Q = \{p_1, p_2, p_3, p_4\}$.



- Let C be the smallest circle that (i) is centered at q, and (ii) covers all the points of Q. Suppose that there is an R-tree on P. Prove that any algorithm using the R-tree to answer the kNN query must access all the nodes whose MBRs intersect C.
- Modify the best first algorithm so that it is guaranteed to access only the nodes whose MBRs intersect C.

Problem 6. Define the L_0 distance between two d-dimensional points p, q as

$$L_0(p, q) = \sum_{i=1}^{d} \Big| p[i] - q[i] \Big|.$$

- Draw the locus of all points in the 2D data space $\mathbb{R}^2$ that have L_0 distance 1 from the origin $(0, 0)$.
- Re-define $mindist(q, r)$ as the smallest L_0 distance from q to the points covered by r. Modify your algorithm in Problem 3 to compute $mindist(q, r)$ in $O(d)$ time.
- Modify the best first algorithm to answer a nearest neighbor query under the L_0 distance optimally.