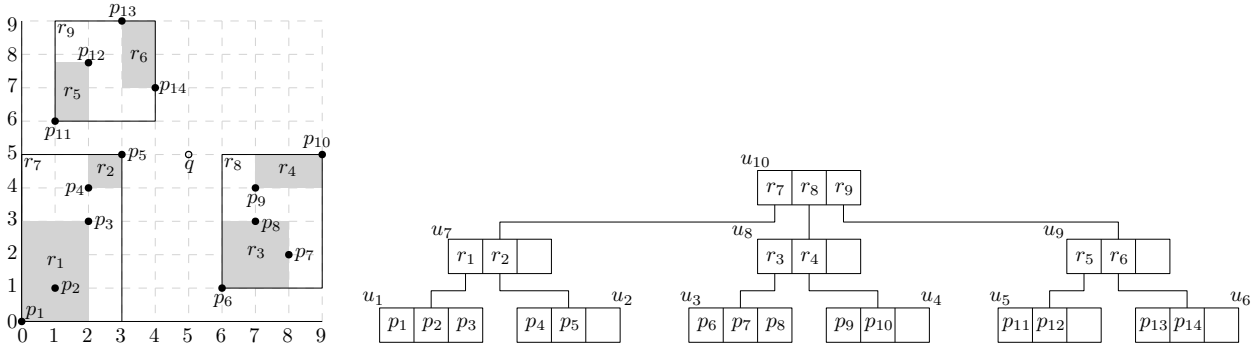
INFS 4205/7205: Exercise Set 3

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.



Solution. $u_{10}, u_8, u_4, u_3, u_9, u_6, u_7, u_2$.

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

Solution. $u_{10}, u_8, u_9, u_7, u_2$.

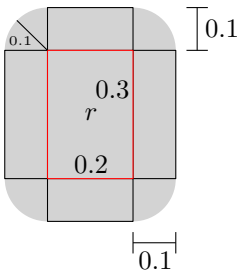
Problem 3. Given an axis-parallel rectangle r and a point q in d-dimensional space, describe an algorithm to compute $mindist(q, r)$ in $O(d)$ time.

Solution. Let $[r_-[i], r_+[i]]$ be the projection of r on dimension $i \in [1, d]$, and $q[i]$ be the coordinate q on dimension i.

Initialize $sqrDist = 0$. For each $i \in [1, d]$: (i) if $q[i] < r_-[i]$, increase $sqrDist$ by $(r_-[i] - q[i])^2$; (ii) if $q[i] > r_+[i]$, increase $sqrDist$ by $(q[i] - r_+[i])^2$; (iii) otherwise, do nothing. After processing all the d dimensions, return $mindist(q, r) = \sqrt{sqrDist}$.

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, $mindist(q, r) \leq 0.1$)?

Solution. C intersects r if and only if q falls in the gray region shown below:

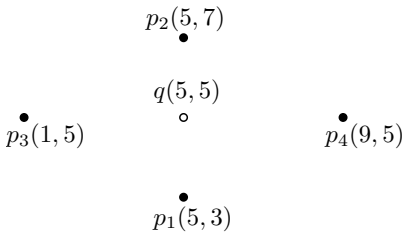


As a result, the probability equals the area of the gray region, which is $0.2 \cdot 0.3 + 2 \cdot 0.1 \cdot 0.2 + 2 \cdot 0.1 \cdot 0.3 + 0.1^2\pi = 0.16 + 0.01\pi$.

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.

Solution. Let us first prove the first bullet. Suppose that this is not true: there exists an algorithm such that it terminates without accessing a node u whose MBR intersects C. Then, it cannot exclude the possibility that u covers a data point falling in C—this point, if exists, must be reported. This contradicts with the fact that the algorithm is correct.

Next, we modify the BF algorithm for the second bullet. In the modified algorithm, we maintain two data structures:

- a sorted list H, where each entry in H is either an MBR whose sorting key is its *mindist* to q, or a point whose sorting key is its distance to q; and
- a set Q of points.

The modified algorithm works as follows. First, initialize $Q = \emptyset$, and insert the MBR of the root to H. Let p_{last} record the the most recent point added to Q; p_{last} = NULL at the beginning. Next, repeat the following steps:

- Remove the smallest entry e from H.
- If e is an MBR r, then access the child node u of r. If u is a internal (or leaf) node, add all the MBRs (or points, resp.) in u to H.

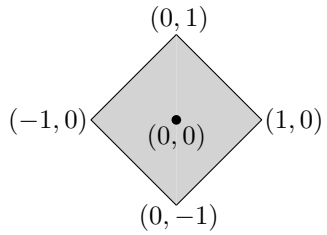
- If e is a point p, then add p to Q, and set $p_{last} = p$.
- If both of the following two conditions hold: (i) $|Q| \geq k$, and (ii) either H is empty or $\|p_{last}, q\|$ is less than the smallest sorting key in H, terminate the algorithm and return Q.

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

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

- 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.

Solution. For the first question, the locus is the region in gray:



Our algorithm for the second question is analogous to the one in Problem 3. Let $[r_-[i], r_+[i]]$ be the projection of r on dimension $i \in [1, d]$, and $q[i]$ the coordinate value of q. Initialize $dist = 0$. For each dimension i: (i) if $q[i] < r_-[i]$, increase $dist$ by $r_-[i] - q[i]$; (ii) if $q[i] > r_+[i]$, increase $dist$ by $q[i] - r_+[i]$; (iii) otherwise, do nothing. After processing all the d dimensions, return $mindist(q, r) = dist$.

For the third question, we can simply run the BF algorithm in exactly the way as discussed in the lecture, replacing Euclidean distance with L_0 distance.