Exercises for CSCI5010

Prepared by Yufei Tao

Problem 1. In the lecture, we presented an algorithm for solving the closest pair problem in $O(n \log n)$ expected time. However, our algorithm requires knowing the precise value of r, which is the distance between the closest pair found from recursion. Computing r precisely would require the "square root" operation, which is not an atomic operation of the real-RAM model. In this problem, you will see how this issue can be circumvented.

(a) In the lecture's algorithm, we imposed a grid where each cell has side length $r/\sqrt{2}$. Suppose that we instead impose a grid whose side length is $c \cdot r/\sqrt{2}$ for some positive constant $c < 1$. Explain how the algorithm can be modified to still find the closest pair correctly in $O(n \log n)$ expected time.

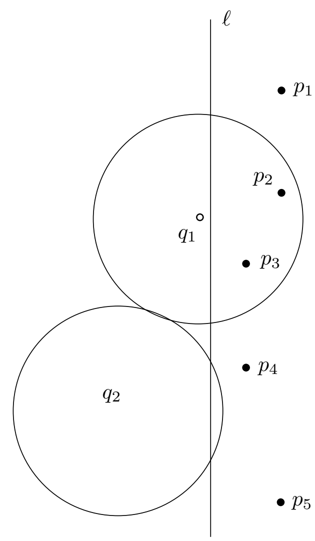
(b) Let p and q be two points whose Euclidean distance is $dist(p, q)$. Given the coordinates of p and q, explain how to obtain in $O(1)$ time a value r' satisfying $dist(p, q)/\sqrt{2} \leq r' \leq dist(p, q)$.

(c) Now modify the closest-pair algorithm in the class to allow a real-RAM implementation that runs in $O(n \log n)$ time.

Problem 2. Design an algorithm that solves the closest pair problem in $\mathbb{R}^d$ in $O(n \log n)$ expected time.

Problem 3*. Let ℓ be a vertical line, and P be a set of n points on the right of ℓ. Define r as the distance of the closest pair of P. It is known that every point in P has distance at most r from ℓ.

We are now given a point q on the left of ℓ. Denote by $D_q(r)$ the disc that centers at q and has radius r. Define an *r-bounded nearest neighbor* (NN) of q as a point $p \in P \cap D_q(r)$ having the smallest distance to q.



For example, in the above figure, $P = \{p_1, p_2, ..., p_5\}$, and r is the distance of p_2 and p_3. The (only) r-bounded NN of q_1 is p_3, whereas q_2 has no r-bounded NNs. The two circles illustrate $D_{q_1}(r)$ and $D_{q_2}(r)$.

Consider the following approach for finding an r-bounded NN of q. First, sort $P \cup \{q\}$ by y-coordinate. Then, inspect the 20 points positioned before and after q in the sorted list, respectively; namely, 40 points are inspected in total. Prove that all r-bounded NNs (if they exist) of q must be among those 40 points.

Hint: Impose a grid, and the constant 40 is rather conservative.

Problem 4*. Let P be a set of points in $\mathbb{R}^2$. Give an algorithm to find the closest pair of P in $O(n \log n)$ time *deterministically*.

Hint: Use the finding in Problem 3.

Problem 5*. Let P be a set of points in $\mathbb{R}^d$ where the dimensionality d is a constant. Give an algorithm to find the closest pair of P in $O(n \log n)$ time *deterministically*.

Hint: How do you generalize your algorithm for Problem 4?