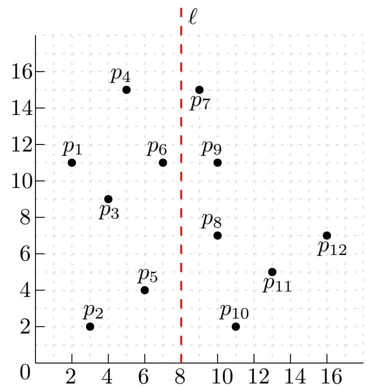
CSCI 5010: Exercise List 12

Problem 1. Consider the set P of points as shown in the figure. Suppose that we run the closest pair algorithm on P. Recall that the algorithm first divides P in halves along the x-dimension using a vertical line ℓ (see the figure), recursively solves each half, and then builds a grid. Answer the following questions:



1. Draw the grid in the figure.

2. Consider the cell c_1 of the grid that covers point p_6. Recall that the algorithm needs to pair up c_1 with certain cells c_2 on the right of ℓ, in order to compute the distance of (p, q) for every pair of points p, q covered by c_1 and c_2, respectively. List the center coordiantes of all such cells c_2.

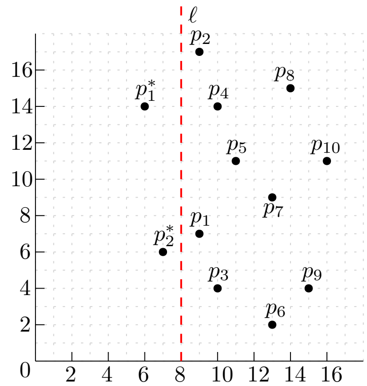
Problem 2. Let P be a set of points in $\mathbb{R}^d$. Give an $O(n \log n)$ expected time algorithm to find the 2nd closest pair of P. Formally, define $T = \{\{p, q\} \mid p, q \in P \wedge p \neq q\}$. The 2nd closest pair is the $\{p, q\} \in T$ that has the second smallest $dist(p, q)$ (i.e., Euclidean distance between p, q).

For instance, in the example dataset Problem 1, the 2nd closest pair is (p_6, p_9) (note that the first closest pair is (p_1, p_3)).

Problem 3. Let ℓ be a vertical line. Let p be a point on the left of ℓ, and P be a set of points on the right of ℓ. Define r as the distance of the closest pair of P. We throw away from P all the points whose distances to ℓ are greater than r. Define P' to be the set of remaining points in P.

For p, we define its *r-bounded nearest neighbor* (NN) as the point q in P that is closest to p, among all the points whose distances to p are at most r (if no such points exist, then p has no r-nearest neighbor).

For example, in the figure below, the closest pair in $P = \{p_1, \ldots, p_{10}\}$ is (p_5, p_7) whose distance is $2\sqrt{2}$. Thus, $r = 2\sqrt{2}$ and $P' = \{p_1, p_2, p_3, p_4\}$. If $p = p_1^*$, then p has no r-bounded NNs, while if $p = p_2^*$, the r-bounded NN of p is p_1.



Consider the following approach of finding the r-bounded NN of p. First, sort $P' \cup \{p\}$ by y-coordinate. Then, identify the position of p in the sorted list. Inspect the 20 points before and after p, respectively (namely, in total 40 points are inspected). Prove that the r-bounded NN (if exists) must be among those 40 points.

Problem 4. Let ℓ be a vertical line. Let P_1 be a set of points on the left of ℓ, and P_2 be a set of points on the right of ℓ. Define r_1 (or r_2) as the distance of the closest pair in P_1 (or P_2, resp.), and $r = \min\{r_1, r_2\}$. Suppose that P_1 and P_2 have been sorted by y-coordinate. Give an $O(n)$ time (where $n = |P_1| + |P_2|$) algorithm to find, for each $p_1 \in P_1$, its r-bounded NN in P_2.

Problem 5. Let P be a set of points in $\mathbb{R}^2$. Give an algorithm to find the closest pair of P in $O(n \log n)$ *worst case* time.