Finding closest points

Bakh Khoussainov

Finding the closest pair of points.

Input: Given n points in the plane.

Output: The pair of points closest to each other.

There is an easy "brute force" algorithm.

Let P be the set of n points:

$$P = \{ p_1, p_2, \ldots, p_n \}$$

$d(x, y)$ = the distance from x to y.

Assumption:

No two points in P have the same x-coordinate or y-coordinate.

We now divide P as follows.

Set-up:

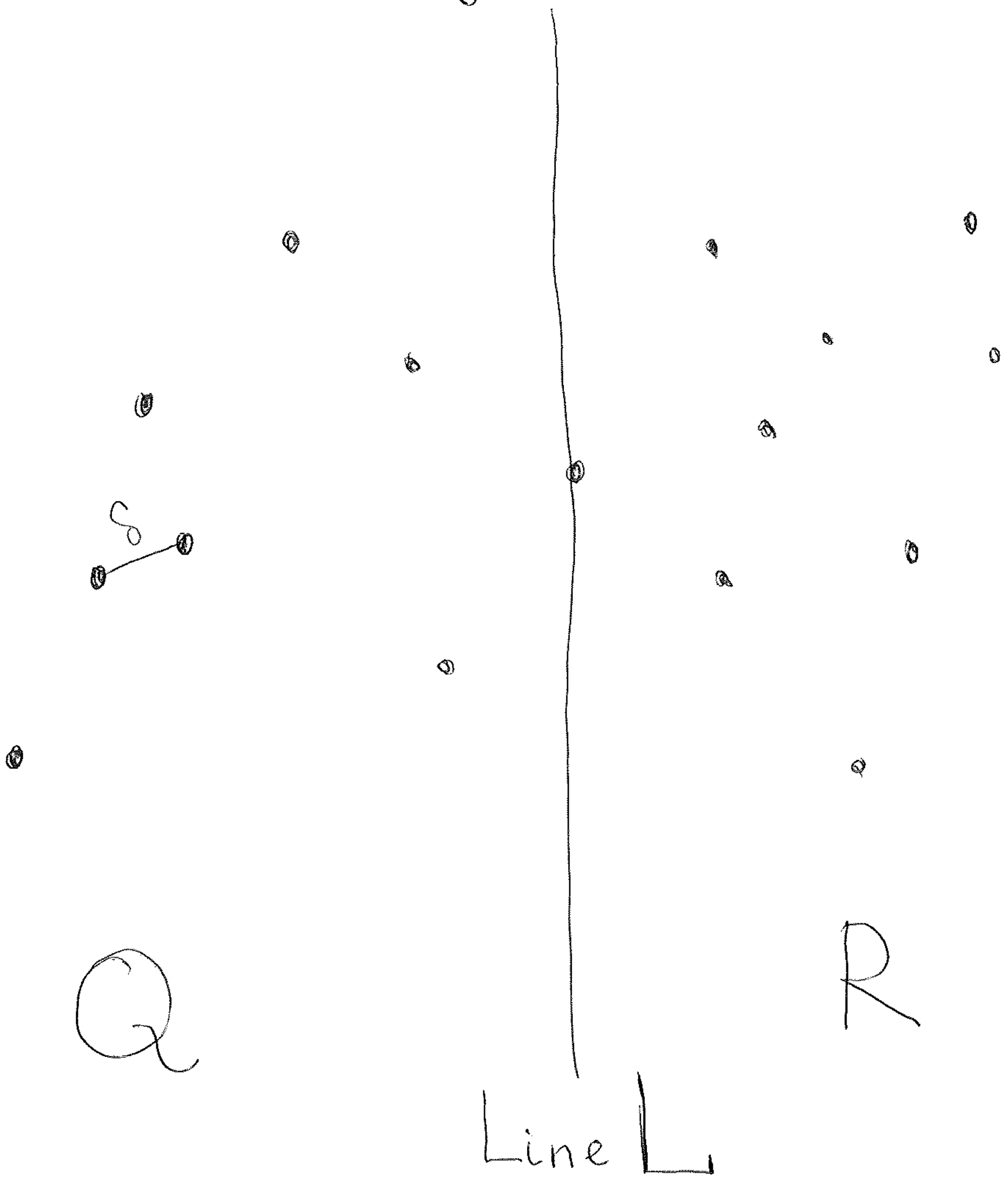
$P_x \leftarrow$ the ordering of P by x-coordinates

$P_y \leftarrow$ the ordering of P by y-coordinates

Q is the set of points in P in the first half positions of P_x.

R is the rest of points.

Pictorially:



Let q_0, q_1 be the closest points in Q.

Let r_0, r_1 be the closest points in R.

Let

$$\delta = \min\{d(q_0, q_1), d(r_0, r_1)\}$$

Question: Are there $q \in Q$, $r \in R$ such that $d(q, r) < \delta$?

Observation. If there are $q \in Q$ and $r \in R$ for which $d(r, q) < \delta$ then both q and r lie within a distance δ of line L,

where L is the line determined by the rightmost x-coordinate of the point in Q. (See picture above).

Indeed, let L be given by the equation $x = x^*$.

Let $q = (q_x, q_y)$, $r = (r_x, r_y)$.

Then

$$x^* - q_x \leq r_x - q_x \leq d(q,r) < \delta$$

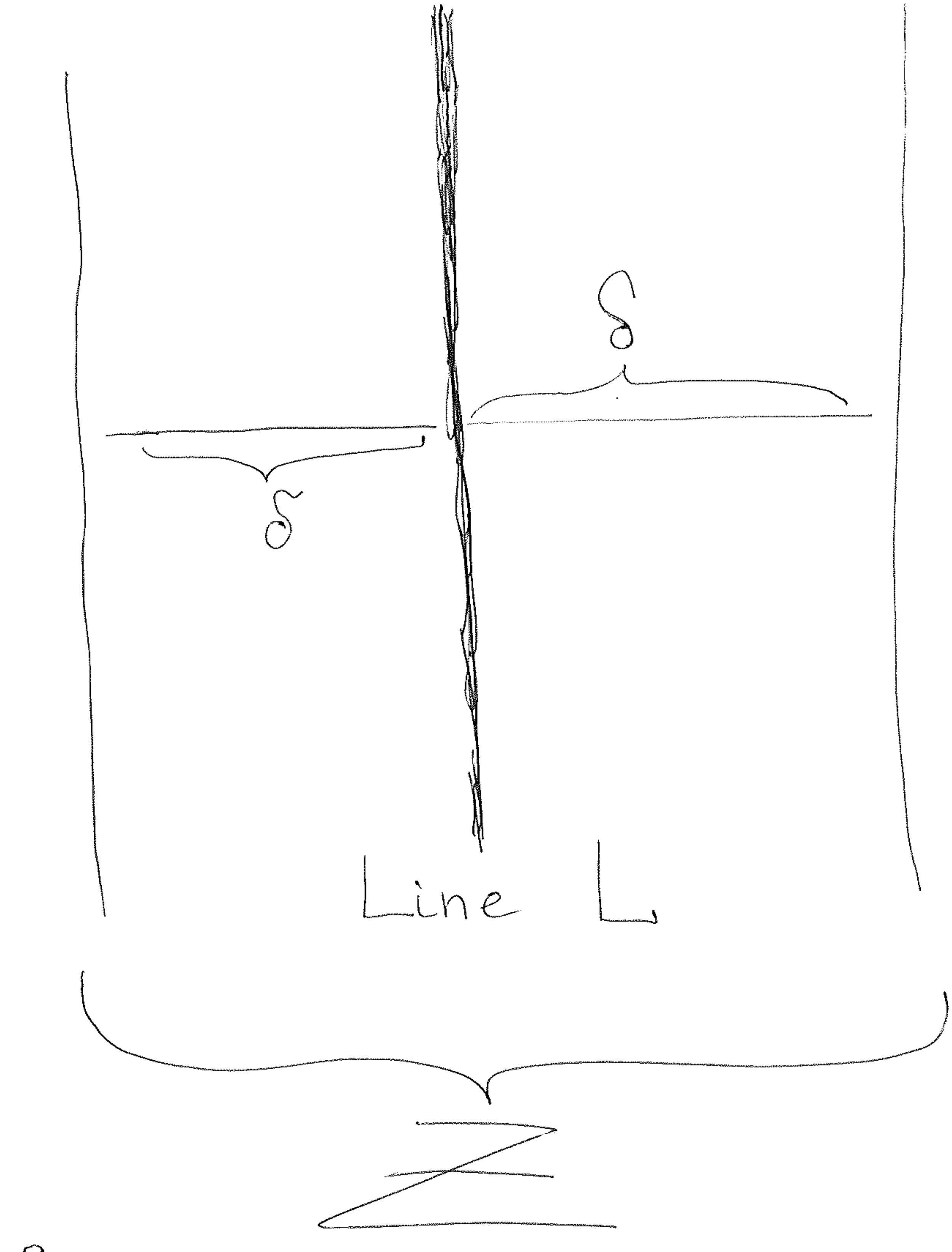
And

$$r_x - x^* \leq r_x - q_x \leq d(q,r) < \delta$$

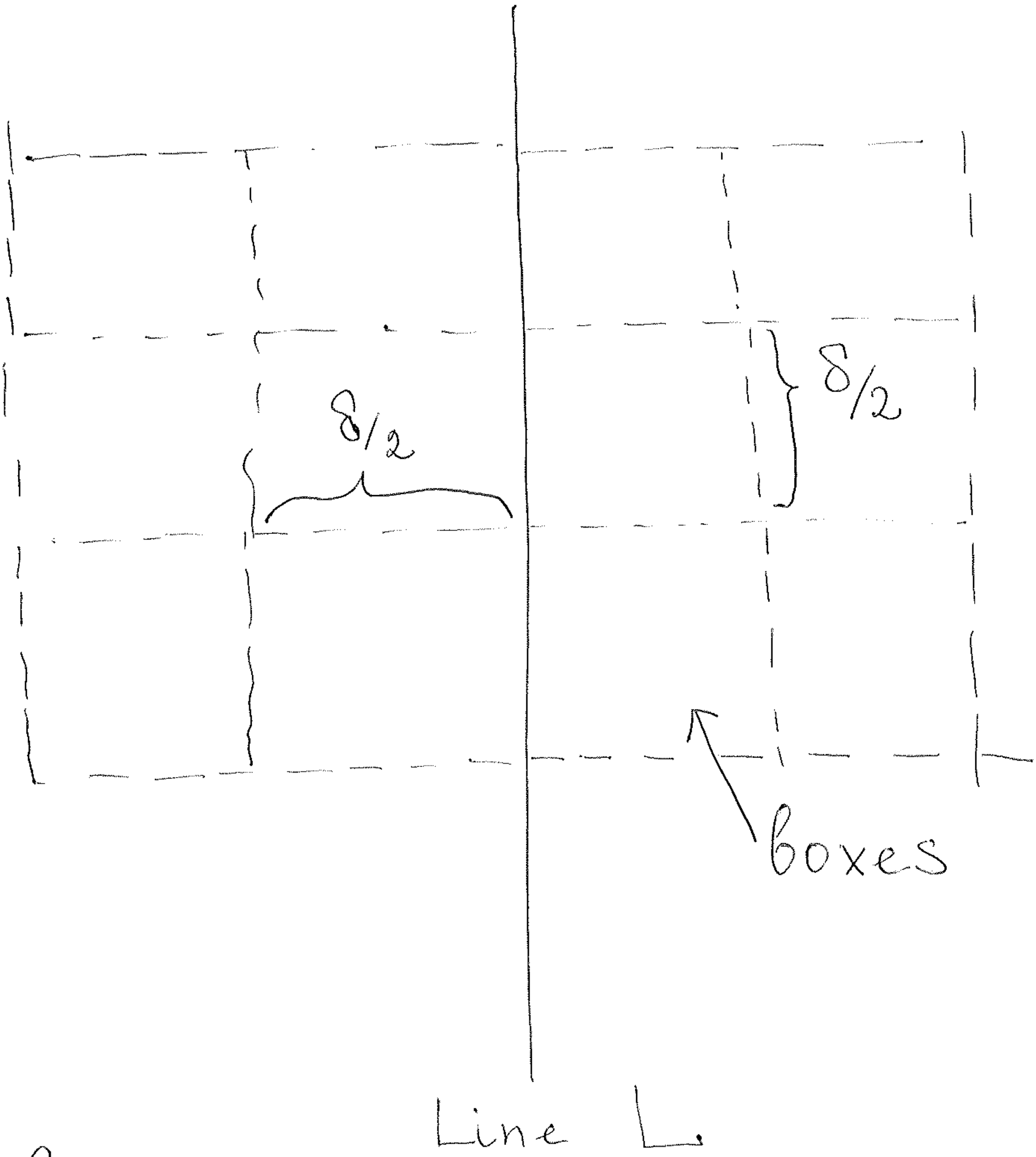
Hence, q and r lie within distance δ of line L.

Consider:

$$Z = \{ p \mid p \text{ within } \delta \text{ of } L \}$$



Partition Z into boxes with horizontal and vertical sides of length $\frac{\delta}{2}$.



Set

$$S = \{ p \in P \mid p \text{ is in } Z \}$$

Claim. Each box contains at most one point of S.

Indeed, if there are two points x, y in one box then $x, y \in Q$ or $x, y \in R$.

Then

$$d(x,y) \leq \sqrt{\frac{\delta^2}{4} + \frac{\delta^2}{4}} = \frac{\sqrt{2}}{2}\delta < \delta.$$

Contradiction.

List S in increasing order of y-coordinates:

S_y.

So, the list S_y is sorted (by y-coordinates).

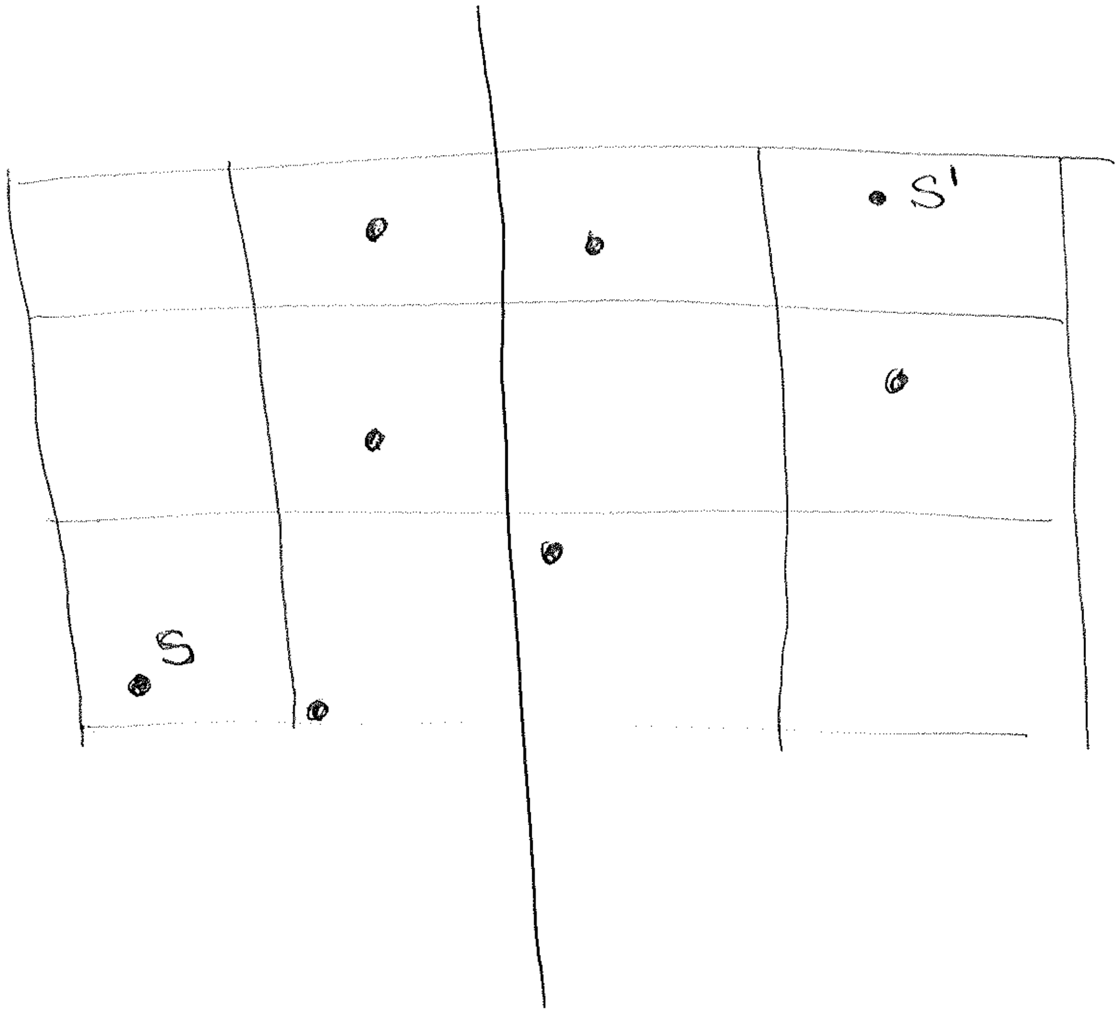
Now assume S has two points $s, s' \in S$ such that

$$d(s,s') < \delta.$$

Claim. In S_y, the points s and s' are within 15 positions.

Indeed, since $d(s,s') < \delta$, there are at most 3 horizontal lines between s and s'. There are at most 15 boxes in that region. Each box contains at most one point of S.

Hence, s and s' are within 15 positions apart in S_y. Pictorially:



$\lrcorner$

The analysis above gives us the following

Closest-Pair (P) algorithm.

Step 1. If P has ≤ 3 points then find the closest pair by measuring all distances.

Step 2. Construct lists P_x and P_y.

Step 3. Construct P, Q.

Let x^* = the max of x-coordinate of a point in Q.

Step 4 (recursive call).

(a) Find the closest pair q_0, q_1 points in Q.

(b) Find the closest pair r_0, r_1 points in R.

Step 5. Set

$$\delta = \min\{d(q_0, q_1), d(p_0, p_1)\}$$

Step 6. Construct

$$S = \{ p \in P \mid p \text{ is within distance } \delta \text{ of } L \}$$

where

$$L = \{ (x,y) \mid x = x^* \}.$$

Step 7. Construct S_y.

Step 8. For $s \in S_y$ compute $d(s,s')$, where s' is within 15 positions from s.

Step 9. Let s and s' be the pair achieving min in Step 8.

If $d(s, s') < \delta$, return (s, s').

Otherwise, return (r_0, r_1) or (q_0, q_1) that gives the distance δ.

The correctness of the algorithm has essentially been proved.

To prove the correctness formally, one needs to use induction on the number of points in P.