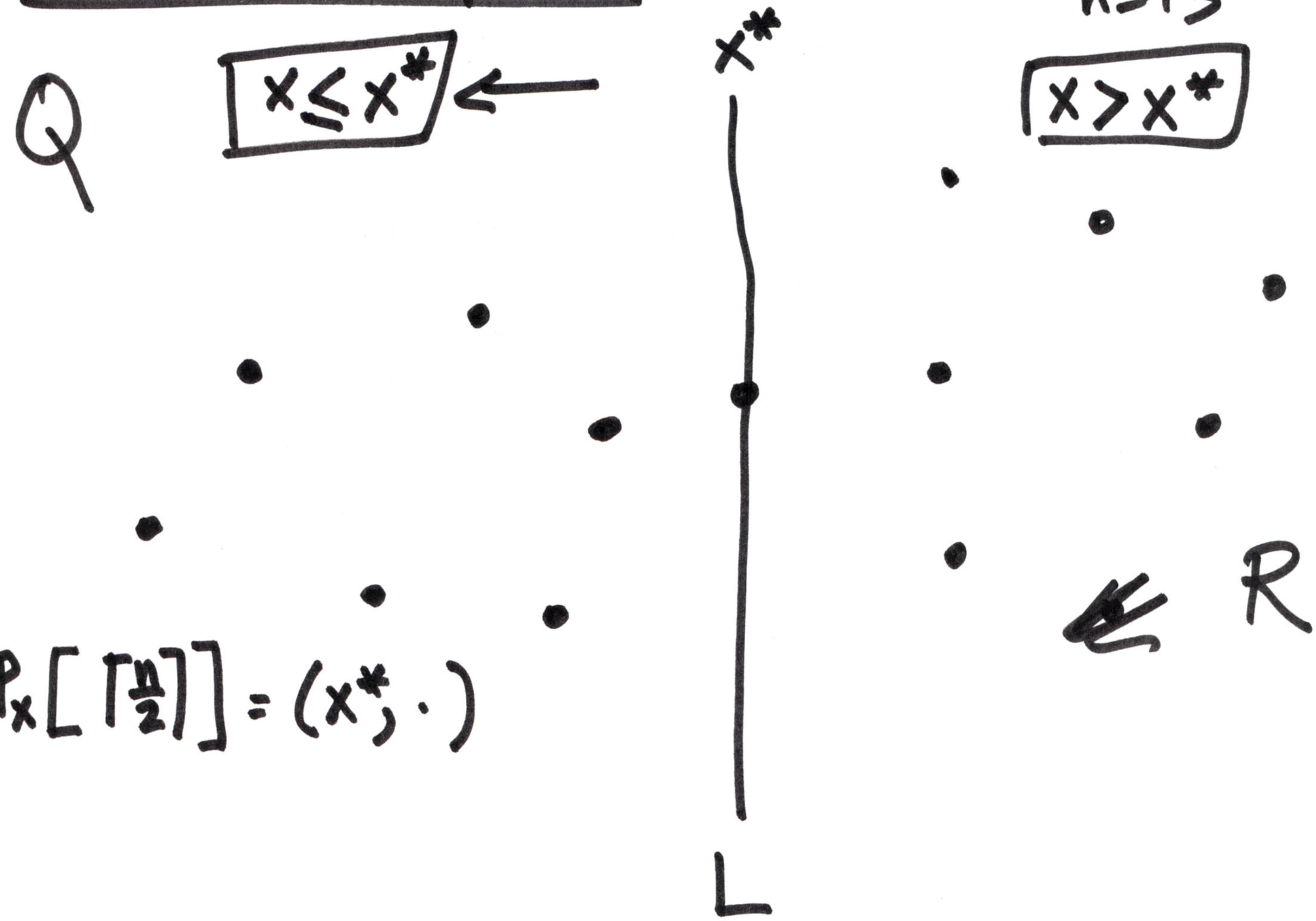
Nov 16, 2011

$P_x \rightarrow P$ sorted by x-coord
$P_y \rightarrow P$ ———— y —— $\Big\} O(n \log n)$

Divide & Conquer



$P_x[\lceil \frac{n}{2} \rceil] = (x^*, \cdot)$

Q_x, Q_y from P_x & P_y in $O(n)$ time
R_x, R_y ————————————

→ Scan thru P_x … $p = (x, y)$

$Q_x \leftarrow p$ $x \le x^*$ $p \rightarrow R_x$

→ Recursively compute closest pairs

$$\delta = \min\left(d(q_1^*, q_2^*),\ d(r_1^*, r_2^*)\right)$$

in Q in R

→ <u>Patch up phase</u>

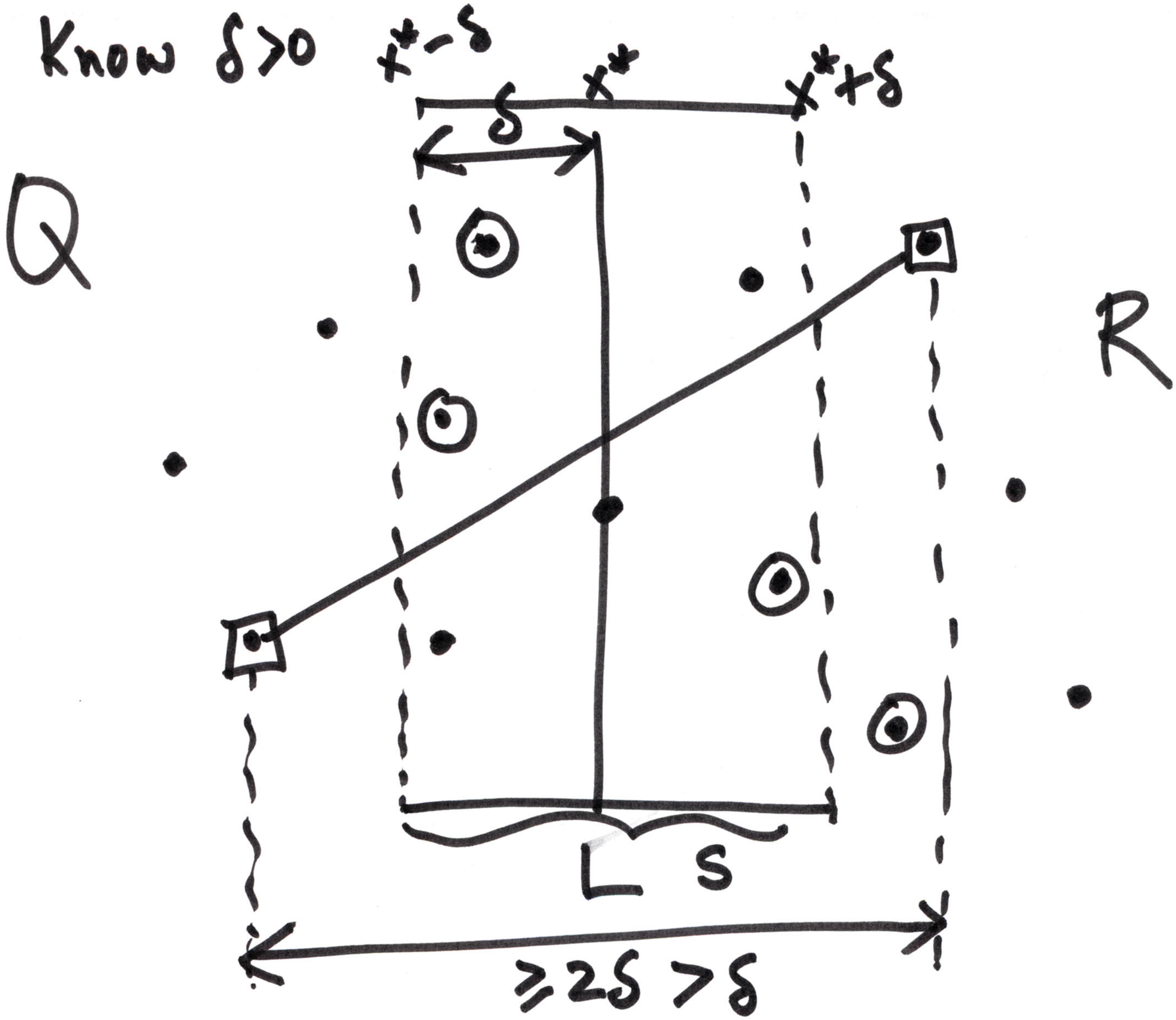
<u>Problem:</u> $\exists\ \underline{q \in Q,\ r \in R}$ s.t. $d(q,r) < \delta$.

If no such pair → done!

otherwise → compute closest crossing pair.

<u>Cool fact</u>: Knowing $\delta \Rightarrow O(n)$ time algo.

Know $\delta > 0$



Lemma: $\exists\, q \in Q,\ r \in R$ s.t. $d(q,r) < \delta$

iff $\exists\, s \neq s' \in S$ s.t. $d(s,s') < \delta$.

Compute S_y from P_y in $O(n)$

S sorted by y co-ord

Kickass Property Lemma:

If $s \neq s' \in S$ s.t $d(s, s') < \delta$ then s & s' are within $\boxed{15}$ positions of each other in S_y.

$\Rightarrow$ $O(n)$ time algo for Closest-in-box

→ Scan through S_y ← n times

$O(1)$ { → for every $p \in S_y$ Compute closest pt within next 15 positions.

Proof idea: