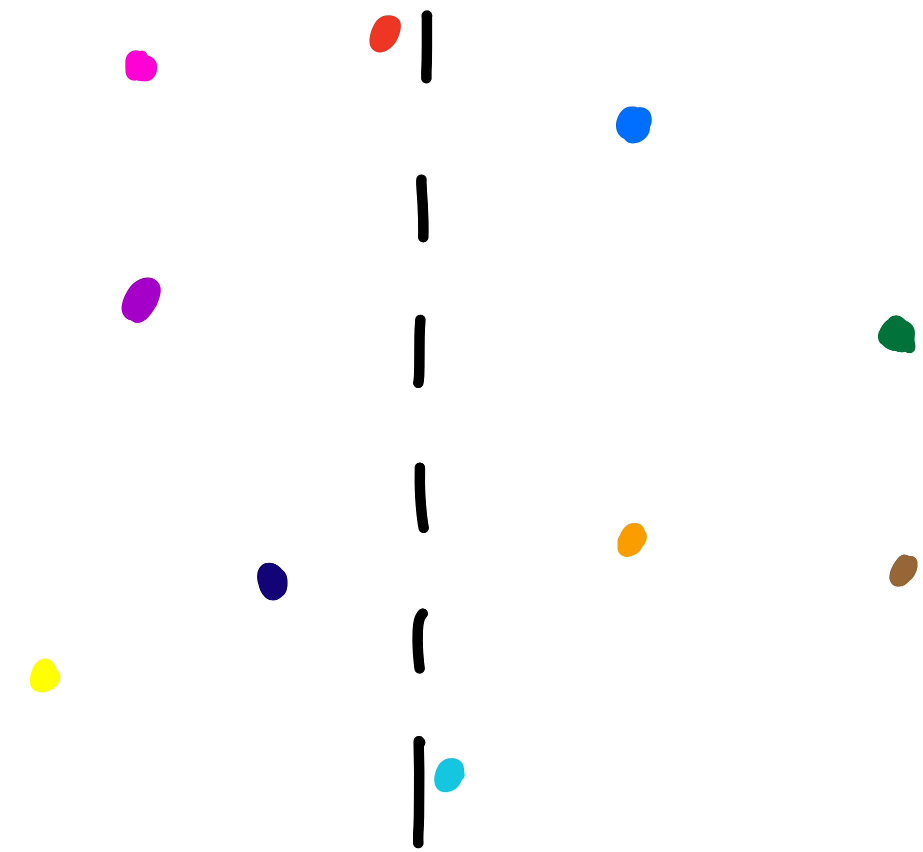
5112

9-12

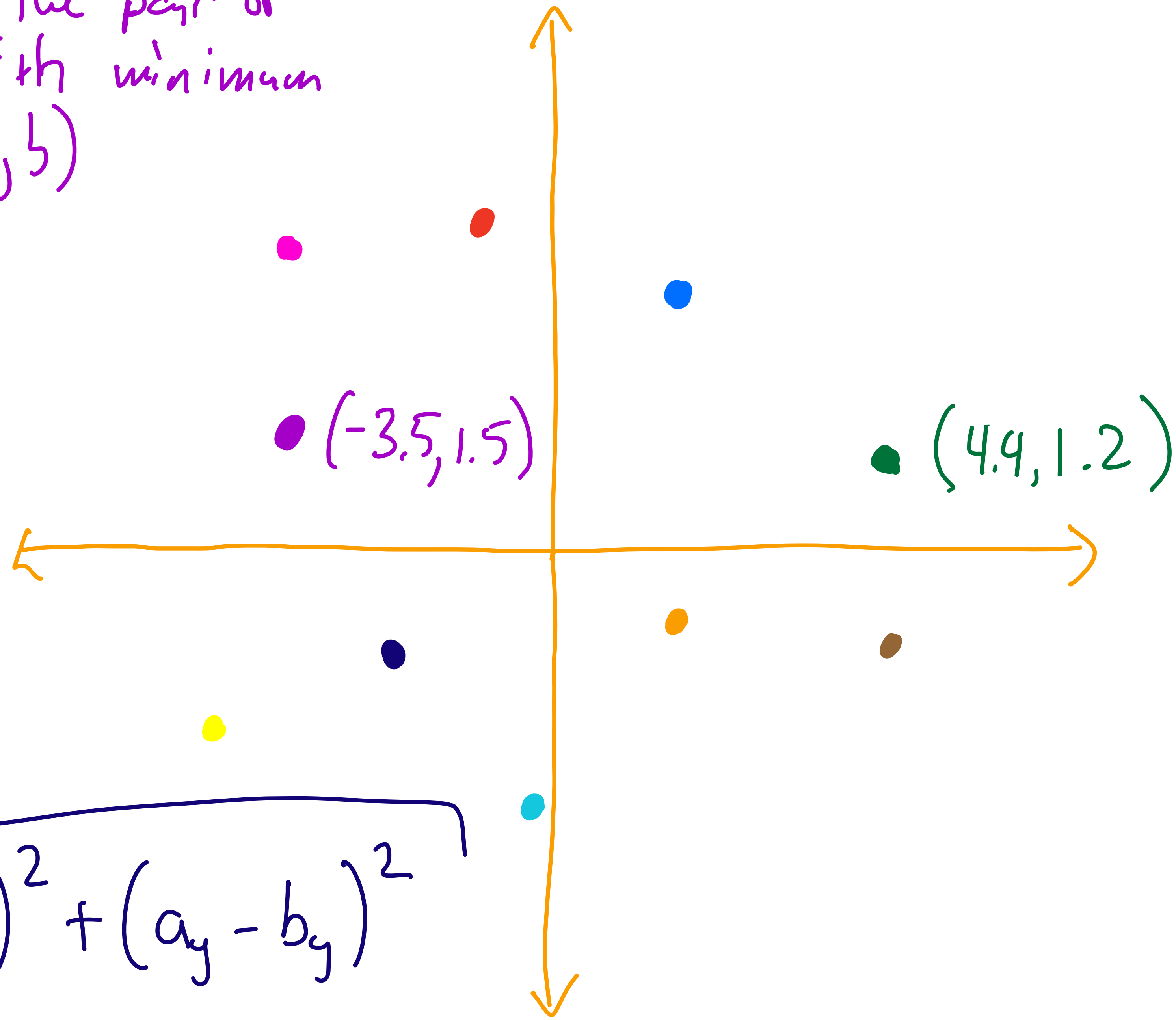
Join Slack

Closest Points on the Plane

Section 5.4



Goal: Return the pair of points a, b with minimum distance d(a, b)



$$d(a,b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

Ideas

1. Could compare all pairs of points - $O(n^2)$

Find a vertical dividing line
(splits the problem in 2)

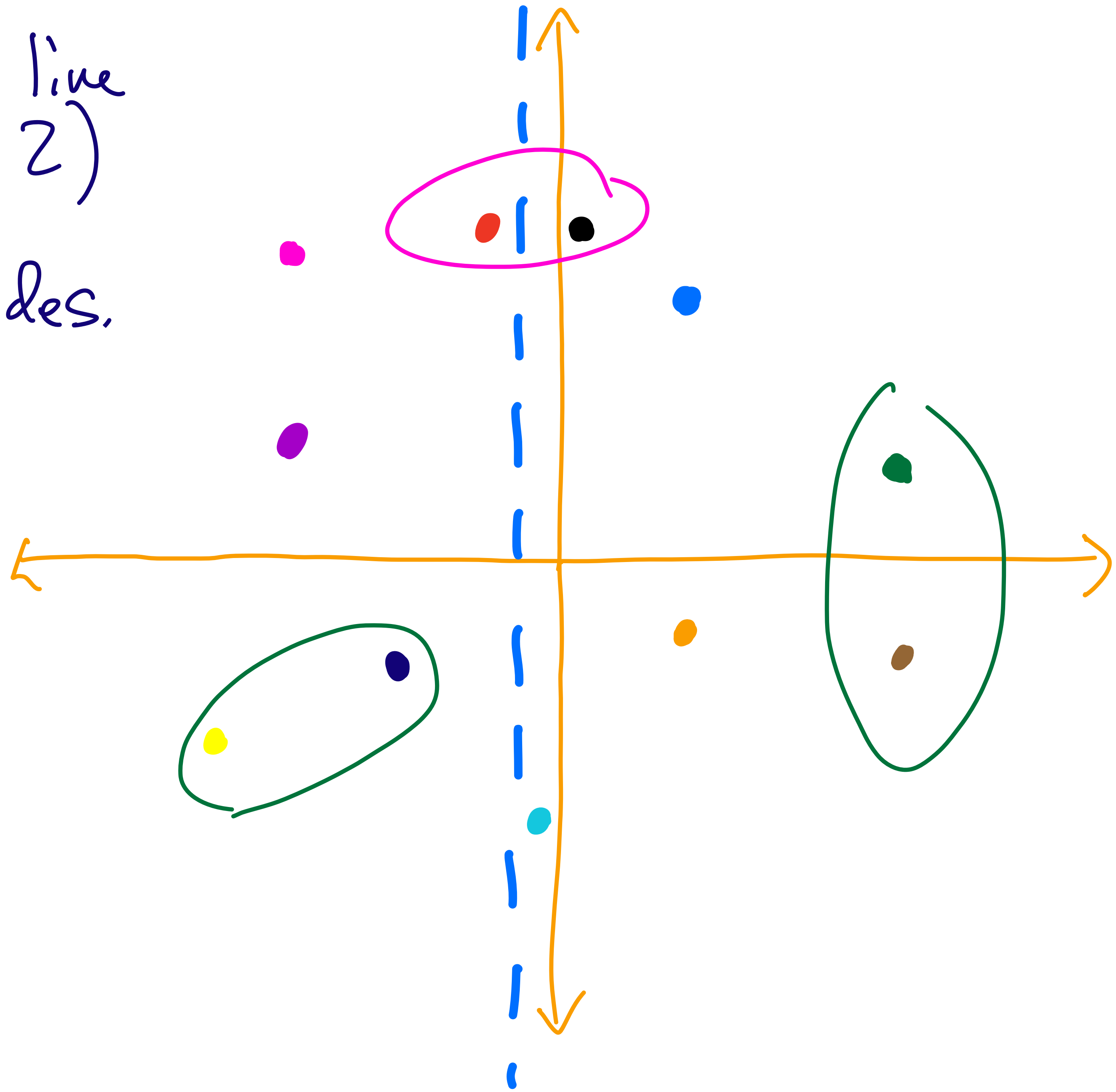
Recursively solve both sides.

Combine the solutions

How to account for pairs that cross the divide?

Try all of them

$n/2 \cdot n/2 = n^2/4$

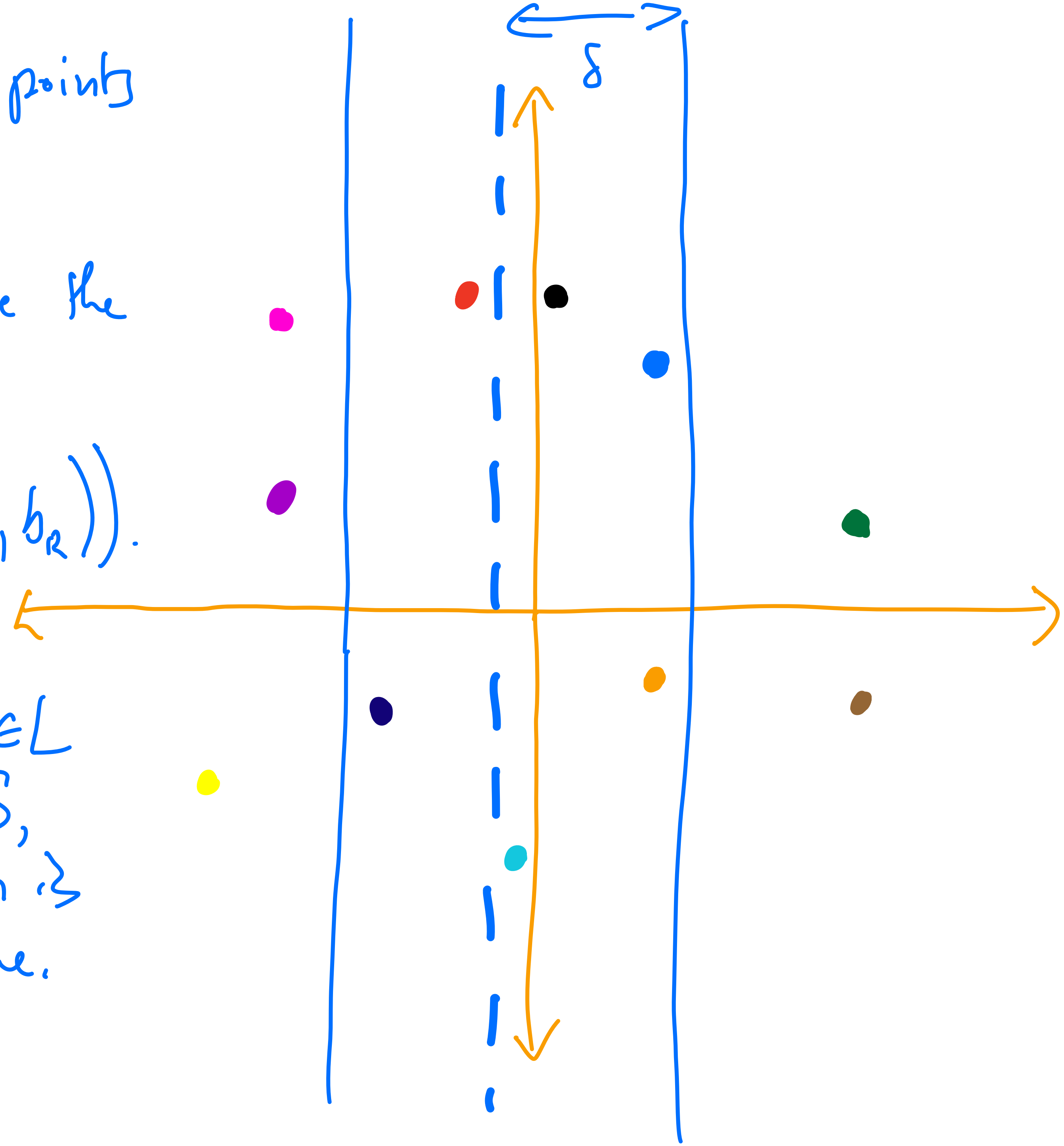


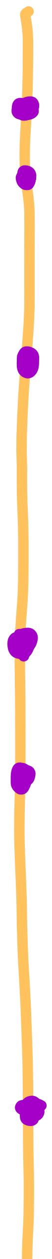
How can we find close points between both sides?

Let a_L, b_L and a_R, b_R be the results of the recursion.

Set $\delta = \min(d(a_L, b_L), d(a_R, b_R))$.

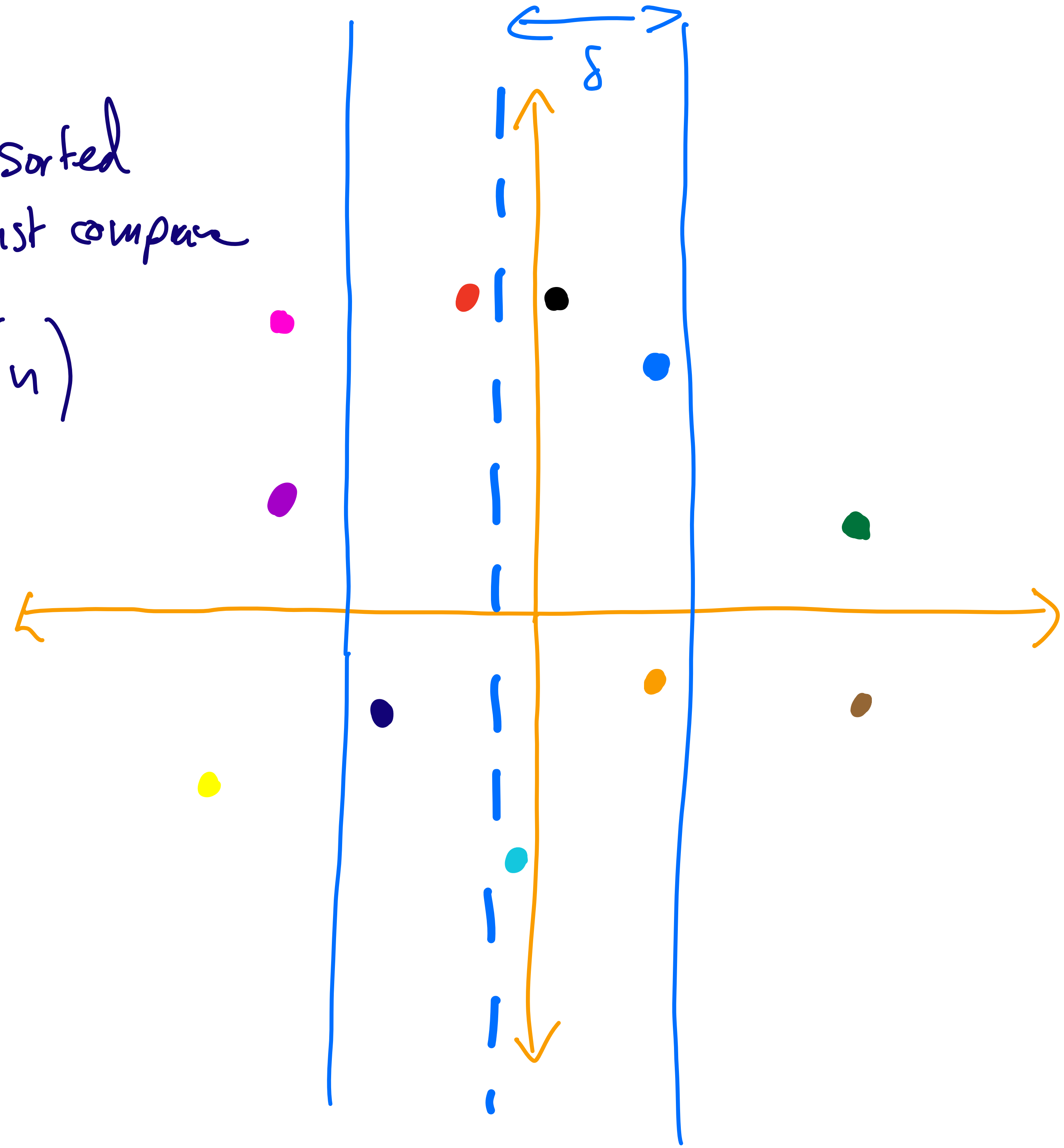
If $d(a,b) < \delta$ for $a \in L$ and $b \in R$, then $a, b \in S$, where S is the strip which is δ around the dividing line.





Start with sorted points, can just compare neighbors.

$O(n)$

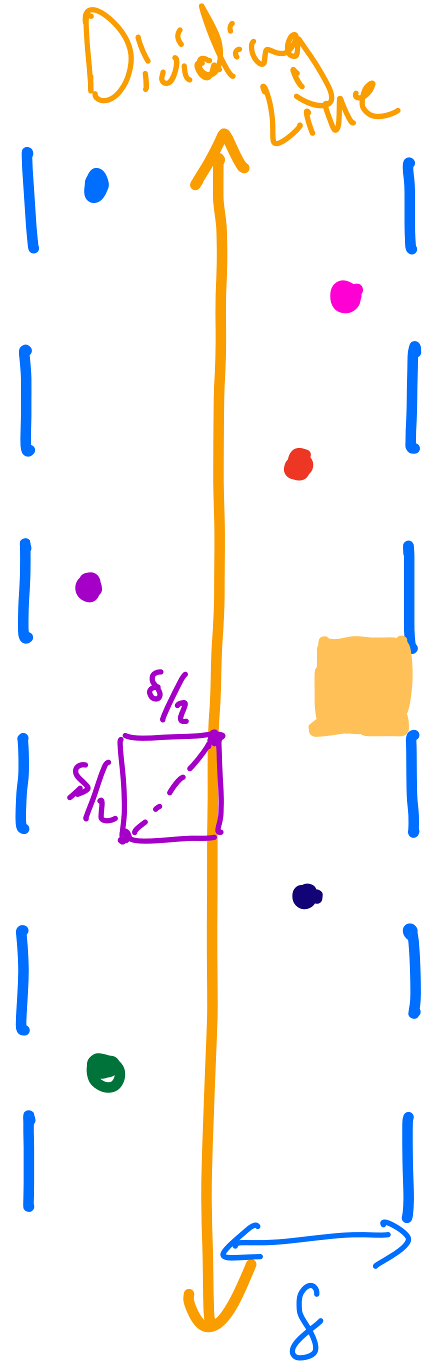


Lemma 1.

No square contains more than 1 point.

Proof. The furthest distance between 2 points in a square is $\sqrt{2}\cdot\frac{\delta}{2} = \delta/\sqrt{2}$.
$< \delta$.

Also 2 points in a square are on the same side. $\Rightarrow$ their distance apart is at least δ. $\square$



Look at a grid of $\delta/2 \times \delta/2$ squares, Lemma 1.

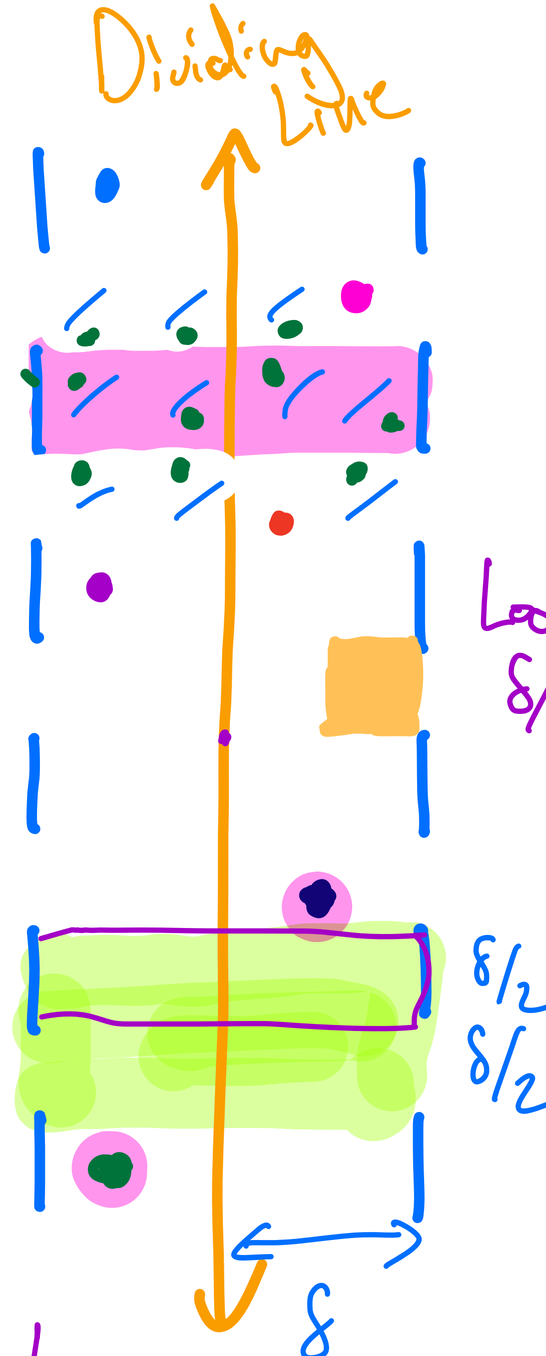
No square contains more than 1 point.

Lemma 2.

Let S_y be the points in S sorted by y-coord. If $a \in L$, $b \in R$ with $d(a,b) < \delta$, then they are within 11 positions of each other in S_y.

Proof. If 2 or more rows separate a and b, then $d(a,b) \geq \delta$. So at most 1 row separates them.

Then $\leq 3+4+3$ points between a and b.
$\overset{\shortparallel}{10}$ ↑ w.r.t. y-coord.



Look at a grid of $\delta/2 \times \delta/2$ squares, Algorithm Input a set of points P.

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$
$a = 2, \quad b = 2$
$f = O(n \log n)$

Start recursion here

Sort P by x-coord $\to P_x$ $O(n \log n)$
Sort P by y-coord $\to P_y$ $O(n \log n)$

$O(n)$ Use P_x to find a dividing line D, get left side L, right side R.

$2T\left(\frac{n}{2}\right)$ Recurse on L and R. ⟸ Filter P_x, P_y to get L_x, L_y and R_x and R_y.
$\to a_L, b_L$ and a_R, b_R, closest pairs in L and R.

$O(1)$ Set $\delta = \min(d(a_L, b_L), d(a_R, b_R))$.
$O(n)$ Set S to be the δ-strip around D.
$\cancel{O(n \log n)}$ Get S_y, S sorted by y-coord. ⟸ Filter from P_y
$O(11n) = O(n)$ Compare each point to the next 11 positions in S_y $\to a_S, b_S$ closest in S
Output the closest pair from a_L, b_L; a_R, b_R and a_S, b_S.

$f(n) = O(n)$

Apply Master Theorem

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

$a = 2, \quad b = 2$

$f = O(n \log n)$

$c_{crit} = \log_b a = 1$

$f(n) \quad n^{c_{crit}}$

$f(n) = O(n^{c_{crit}} \log^k (n)) \quad k = 1$

$\Rightarrow T(n) = O(n^{c_{crit}} \log^{k+1} (n))$

$= O(n \log^2 n)$

Apply Master Theorem Again

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

$a = 2, \quad b = 2$

$f = O(n)$

$c_{crit} = 1$

$f(n) = O(n^{c_{crit}} \log^k (n)) \quad k = 0$

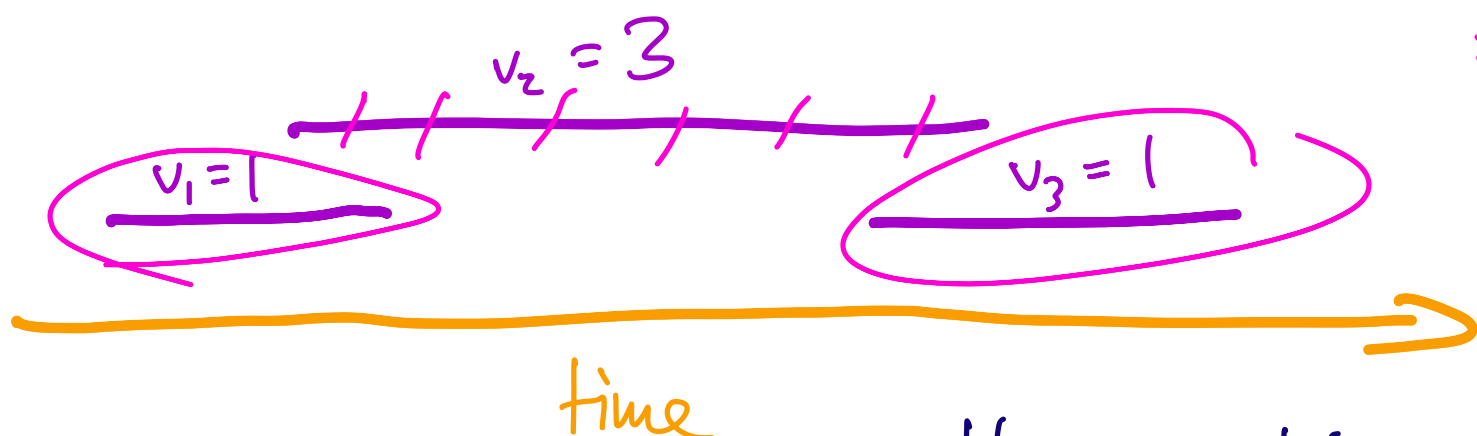
$O(n \log n)$.

Dynamic Programming

Weighted Interval Scheduling

Input : Set of intervals $I_1, I_2, \ldots, I_n$ $\quad I_j = (s_j, f_j)$

v_j value of I_j



EFT $\rightarrow$ output value 2.

Have to take values into consideration.