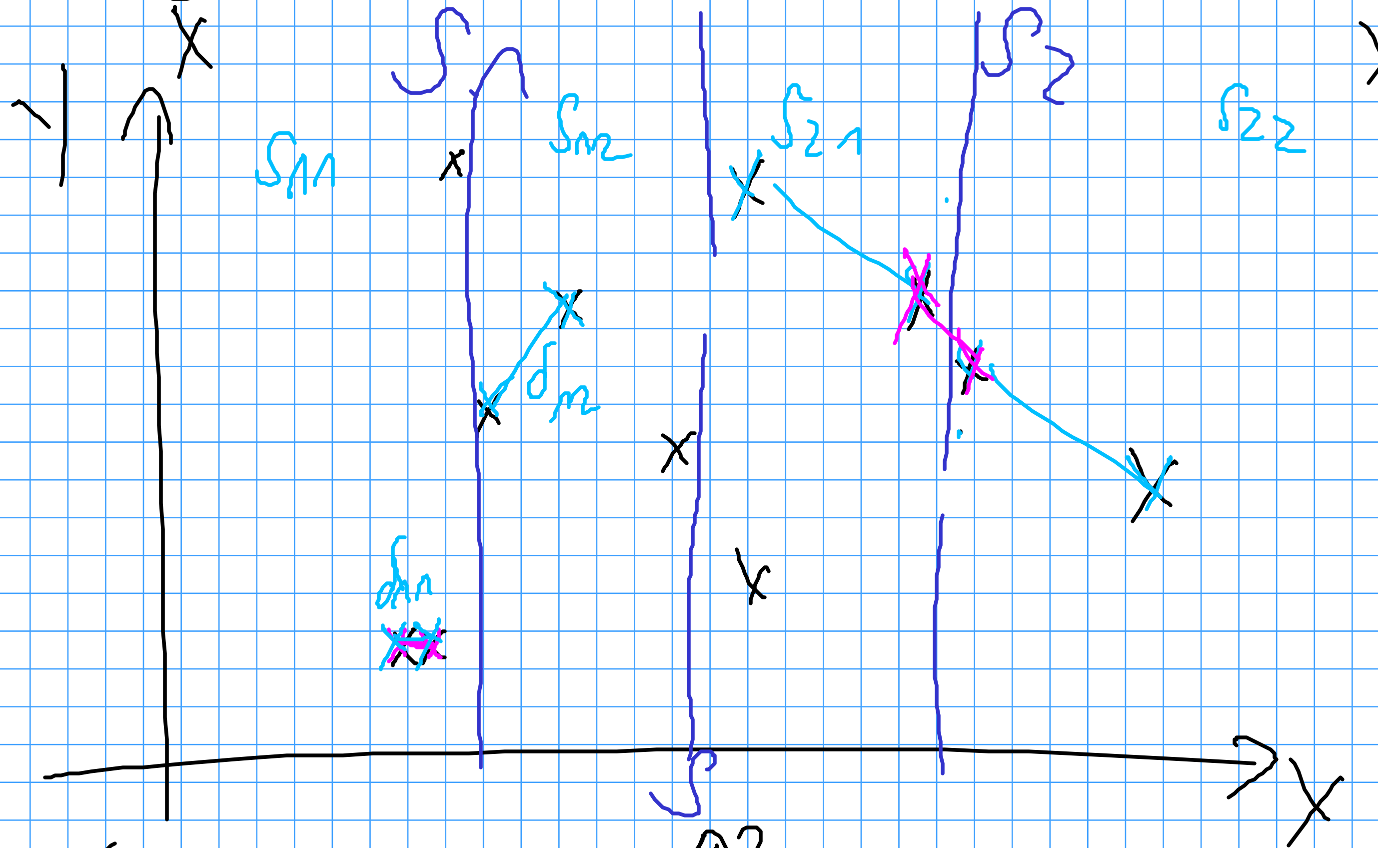
3.1. CLOSEST PAIR PROBLEM



S – set of n points $p \in \mathbb{R}^2$

Which two points have minimal distance?

Naive:

- compute all pair-wise distances, keep track of the minimum

$$O(n^2)$$

1° Divide & Conquer $O(n \log n)$

Divide

1. If $|S| \le 10$, brute force the CP, let the distance be δ,

else go to 2

2. compute x^* – median of all x-coordinates of points in S

S_1 – $p \in S : x(p) < x^*$

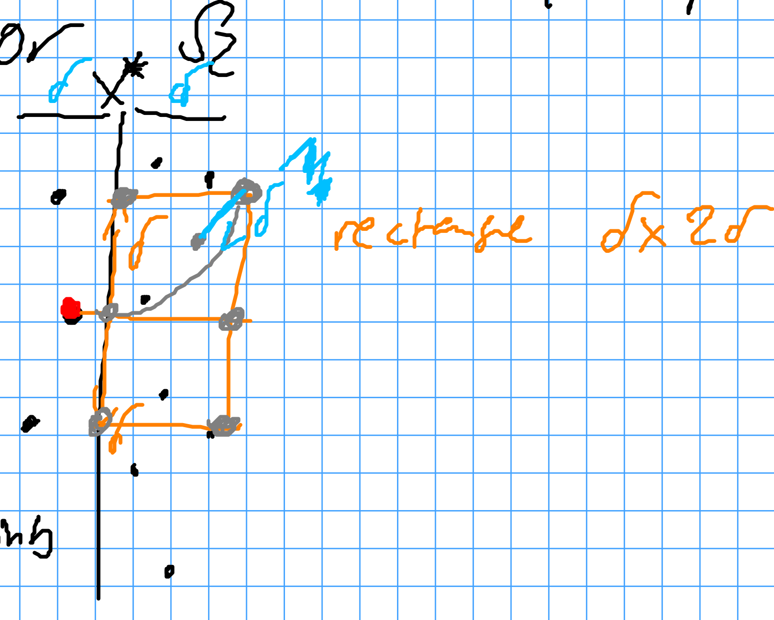
S_2 – $p \in S : x(p) > x^*$

$p \in S : x(p) = x^*$ ⤳ divide them s.t. that $|S_1| = |S_2| = \frac{|S|}{2}$

3. recursively perform step 1 & 2 on S_1, S_2

Conquer:

- δ = min CP distance of a pair in S_1 or S_2



- to get CP of $S = S_1 \cup S_2$ consider all points

in δ - interval
around x^* BUT: could be that
we have all points in
there
$$\frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4}$$

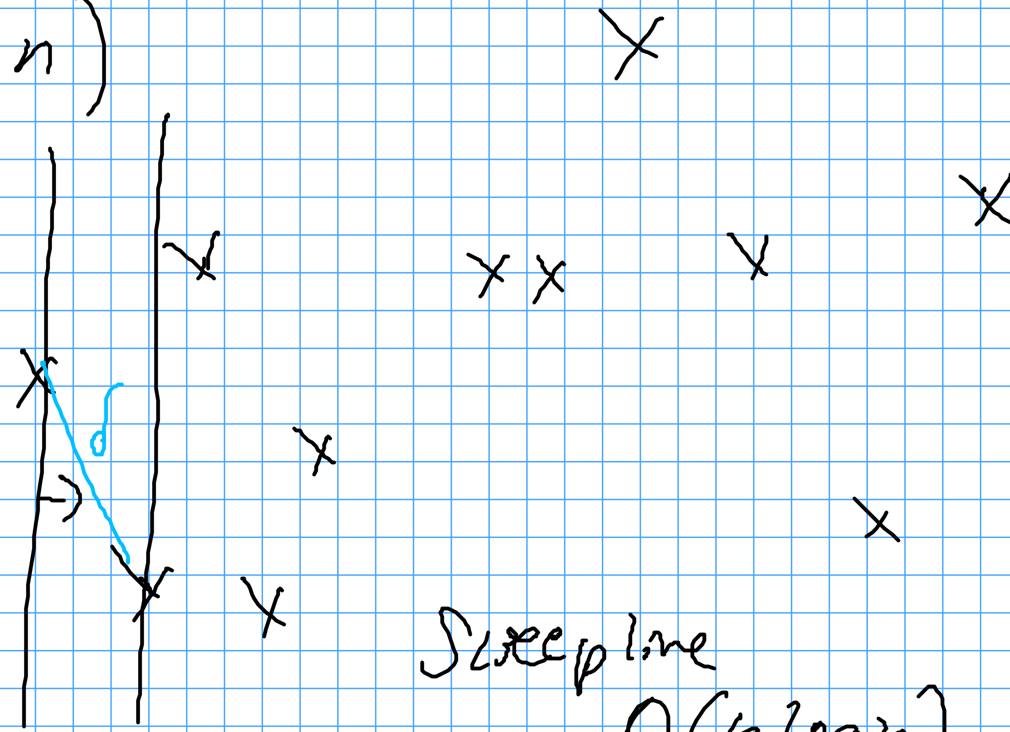
now: consider only points that are
not too far away in terms of
the y - coordinate
$\Rightarrow$ $\delta \times 2\delta$ rectangle
$\Rightarrow$ at most 6 points
in there

$\Rightarrow$ constant number of comp.
per node in S_n

$\Rightarrow$ $O(n)$

$\rightarrow$ for efficient access to the
candidate nodes, we need them
pre-sorted by y - coordinate
↳ on demand sorting takes too
long $O(\frac{n}{2} \log \frac{n}{2})$, but bottom
up merge sort can be
applied

$\Rightarrow O(n \log n)$



Sweepline
$O(n \log n)$

Deterministic Lower Board

↳ Sweepline and Divide & Conq. are asymptotical optimal because there is a matching lower bound

$\Rightarrow$ Element Uniqueness Problem (EU)

$S = \{x_1, \dots, x_n\}$ $x_i \in \mathbb{R}$

are all numbers unique?

$\Rightarrow$ lower bound of $O(n \log n)$

↳ Reduction: EU can be reduced to CP

For every $x_i \in S$, we construct a point $p = (x_i, x_i) \in \mathbb{R}^2$ $\Rightarrow O(n)$

$\hookrightarrow$ invoke CP alg., if CP has a distance of $0 \rightarrow$ EU false

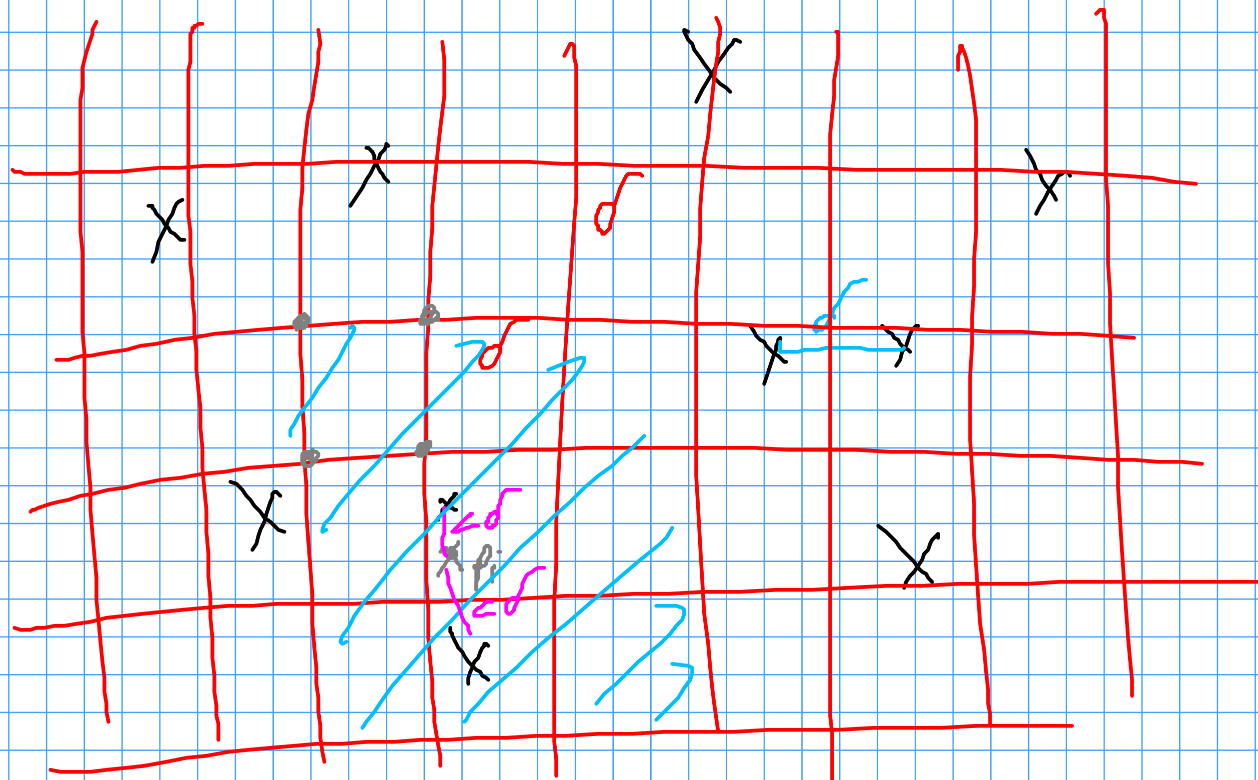
else EU true

$\Rightarrow$ if a det. alg. solves CP faster than $O(n \log n)$, EU could be solved faster $\not\downarrow$

Randomized Incremental Algorithm

runs in $O(n)$

Let $p_1, p_2, \ldots, p_n$ the points in some order. Lets assume for a moment that we know about the CP distance for the points in $p_1, \ldots, p_{i-1}$, we call this distance δ.



build grid layer with cell length δ

sort all $p_1, \dots, p_{i-1}$ in the respective cell $O(i-1)$

Now consider p_i and let C_{ee} be its grid cell, a candidate to form a CP with p_i has to be in C_{ab} $a \in \{e-1, e, e+1\}$
$b \in \{e-1, e, e+1\}$

9 cells

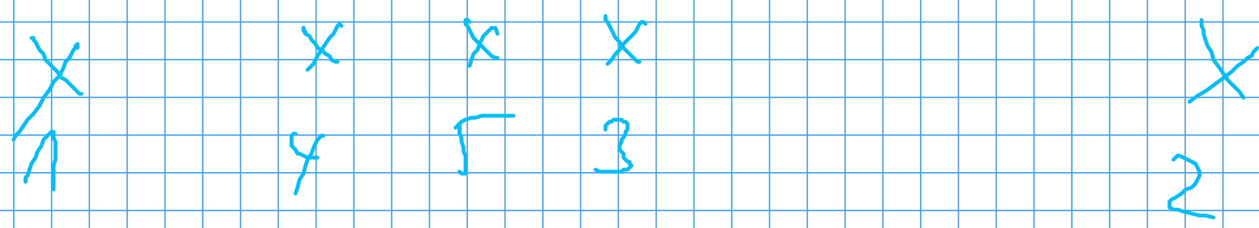
↳ per cell at most 4 points can lie inside, because otherwise their distance would smaller than δ ↳ at most

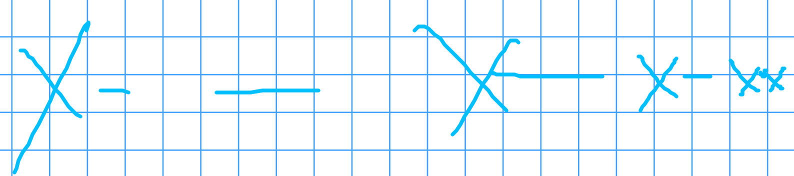
3b comparisons

$\Rightarrow$ total runtime $\sum_{i=3}^{n}(O(i)+O(n))=O(n^2)$

BUT: only have to rebuild grid layer if p_i forms new CP

EXERCISE Fix an instance of CP and an order of the points, such that the grid needs to be rebuild in every iteration.

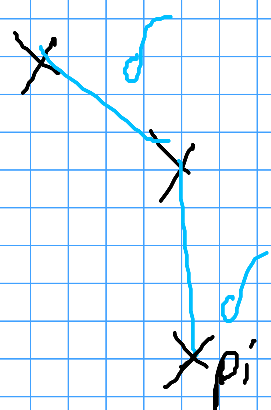




Now consider the points in random order.

EXERCISE If points are in random order, what is the probability of point p_i to form a new CP?

$\leq \frac{2}{i}$ because 2 of the i points form a CP, and with every permutation being equally likely the chance that p_i is one of them is $\frac{2}{i}$



Expected cost per iteration

$$O(i) \cdot \frac{2}{i} + \frac{i-2}{i} \cdot O(1)$$

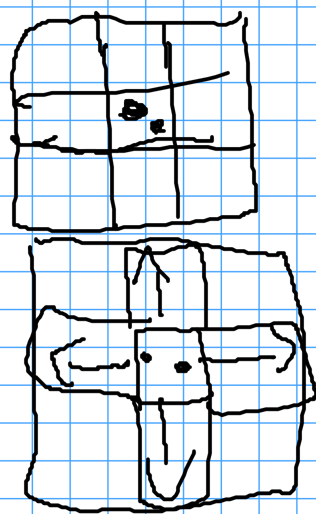
$$= O(1)$$

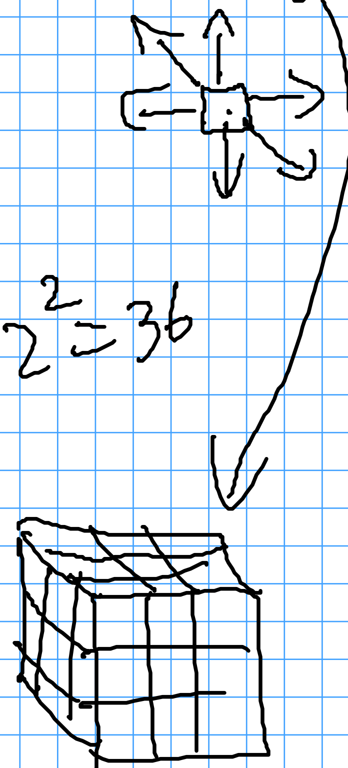
total runtime:

$$T(n) = \sum_{i=3}^{n} O(1) = O(n)$$

EXERCISE What is the runtime of this CP alg. in d dimensions?

$3^3 \cdot 2^3 = 216$





$3^2 \cdot 2^2 = 36$

$$3^D \cdot 2^D = 6^D$$

for fixed D $\rightarrow$ constant