

Lecture 27

CSE 331

Nov 4, 2022

Coding P2 due TODAY

Fri, Nov 4	Kickass Property Lemma     x^2	[KT, Sec 5.4] (Project (Problem 2 Coding) in)
Mon, Nov 7	Weighted Interval Scheduling    x^2	[KT, Sec 6.1] (Project (Problem 2 Reflection) in)
Tue, Nov 8		(HW 6 out)
Wed, Nov 9	Recursive algorithm for weighted interval scheduling problem    x^2	[KT, Sec 6.1]
Fri, Nov 11	Subset sum problem     x^2	[KT, Sec 6.1, 6.2, 6.4]
Mon, Nov 14	Dynamic program for subset sum     x^2	[KT, Sec 6.4]
Tue, Nov 15		(HW 7 out, HW 6 in)
Wed, Nov 16	Shortest path problem     x^2	[KT, Sec 6.8]
Fri, Nov 18	Bellman-Ford algorithm     x^2	[KT, Sec 6.8]
Mon, Nov 21	The P vs. NP problem  	[KT, Sec 8.1]
Wed, Nov 23	No class	Fall Recess
Fri, Nov 25	No class	Fall Recess
Mon, Nov 28	More on reductions  	[KT, Sec 8.1]
Tue, Nov 29		(HW 8 out, HW 7 in)
Wed, Nov 30	The SAT problem  	[KT, Sec 8.2]
Fri, Dec 2	NP-Completeness  	[KT, Sec. 8.3, 8.4] (Project (Problem 3 Coding) in)
Mon, Dec 5	k -coloring problem  	[KT, Sec 8.7] (Quiz 2) (Project (Problem 3 Reflection) in)

Group formation instructions

Autolab group submission for CSE 331 Project

The lowdown on submitting your [project](#) (especially the [coding](#) and [reflection](#)) problems as a group on Autolab.

Follow instructions **EXACTLY** as they are stated

The instructions below are for Coding Problem 1

You will have to repeat the instructions below for EACH coding AND reflection problem on project on Autolab (with the appropriate changes to the actual problem).

Form your group on Autolab

Groups on Autolab will NOT be automatically created

You will have to form a group on Autolab by yourself (as a group). Read on for instructions on how to go about this.

Make sure you are in your group

note #386   

stop following **2 views** [Actions](#)

Coding P1 due today

A gentle reminder that the [first coding problem](#) is due by 11:58pm tonight!

Finally, make sure that you are officially included in your group on Autolab for the coding problem 1 before your group submits its code. If you are not included in the group on Autolab, you will get a ZERO on coding problem 1.

Please make sure that you verify that you see a submission for yourself on Autolab. It is your **PERSONAL RESPONSIBILITY** to make sure that this is the case. If your group forgets to do this is it your responsibility to remind them that you need to be included.

If your group has already submitted without you, make sure you are included in the group on Autolab and then someone from your group should re-submit.

project

[Edit](#) good note 

Updated 2 minutes ago by Art Rude

Coding problem grading

note @391 stop following 38 views Actions

Please make sure you read the coding grading rubric correctly

Looks like few of your are not interpreting how the grading for problems 2-5 works (@383).

Please note that if your revenue exactly matches the revenue of the optimal solution for problem 1 you should be getting level 0. Note that the ratio that determines your level is

$$\frac{\text{(your solution's revenue - revenue of optimal Solution for Problem 1)}}{\text{(our revenue - revenue from optimal Solution for Problem 1)}}$$

If you revenue matches that of optimal solution for problem 1, your case the numerator is 0 and hence your level will be 0.

project

Edit good note | 1

Updated 12 hours ago by Atri Rudra

OH today shortened to 30 mins

note @380    stop following **1 view** Actions

My Friday Office hours will be for 30 minutes

So sorry to do this but my Friday OH for Nov 4 will be for 30 mins from 12:45-1:15pm. *This change is only for this week and the Wed OHs times will not change.*

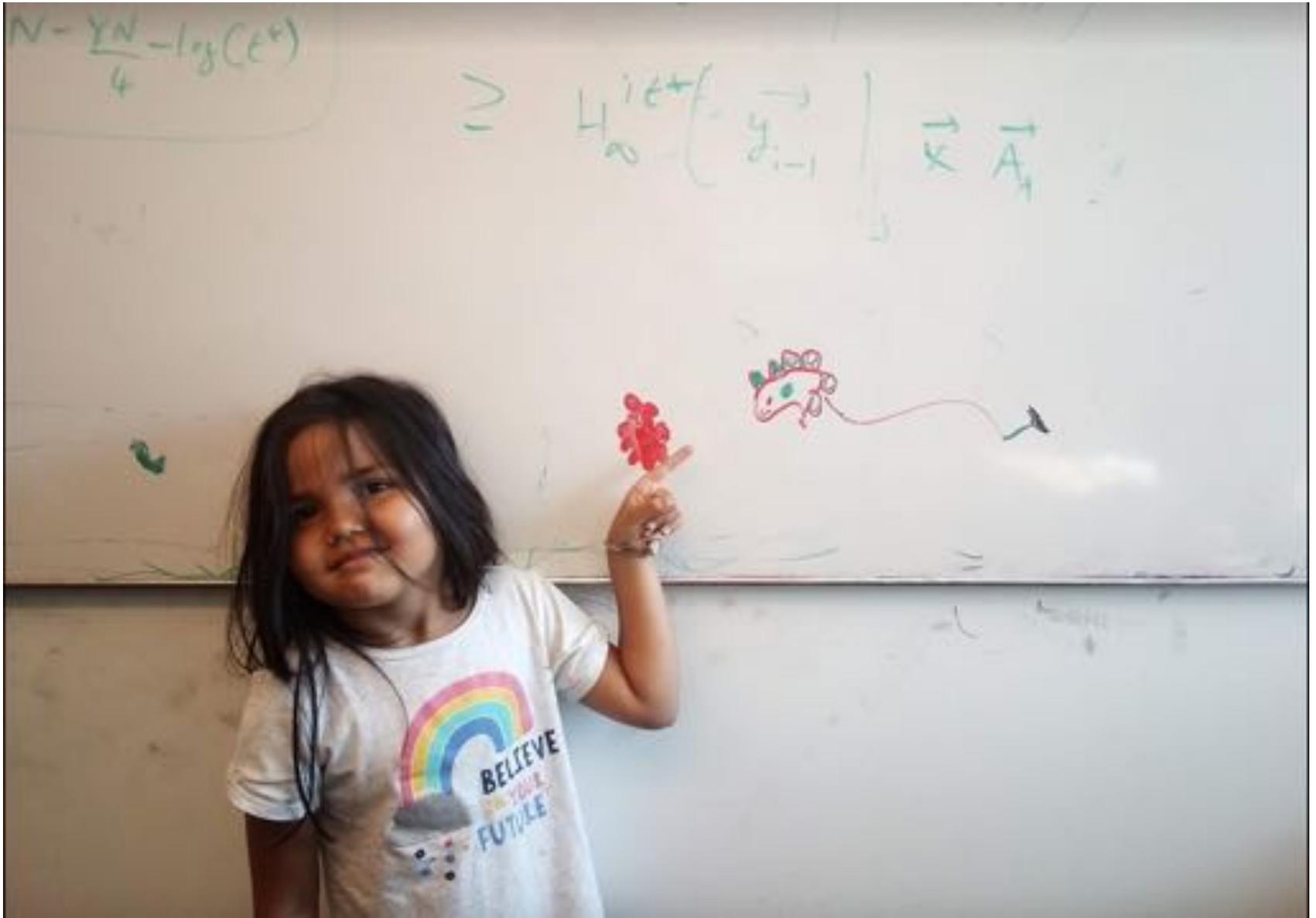
office_hours

Edit good note | 0 Updated 31 seconds ago by Atri Rudra

Have fun @ UB Hacking!



Questions/Comments?

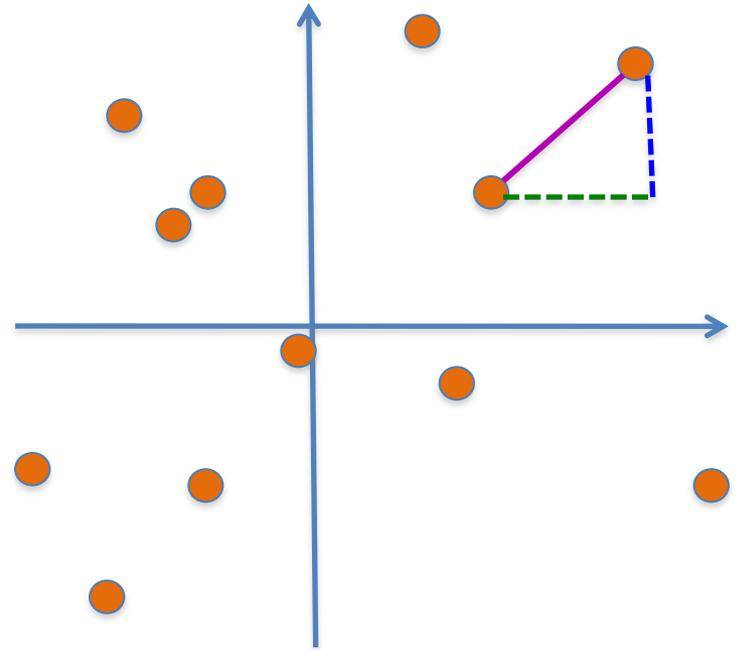


Closest pairs of points

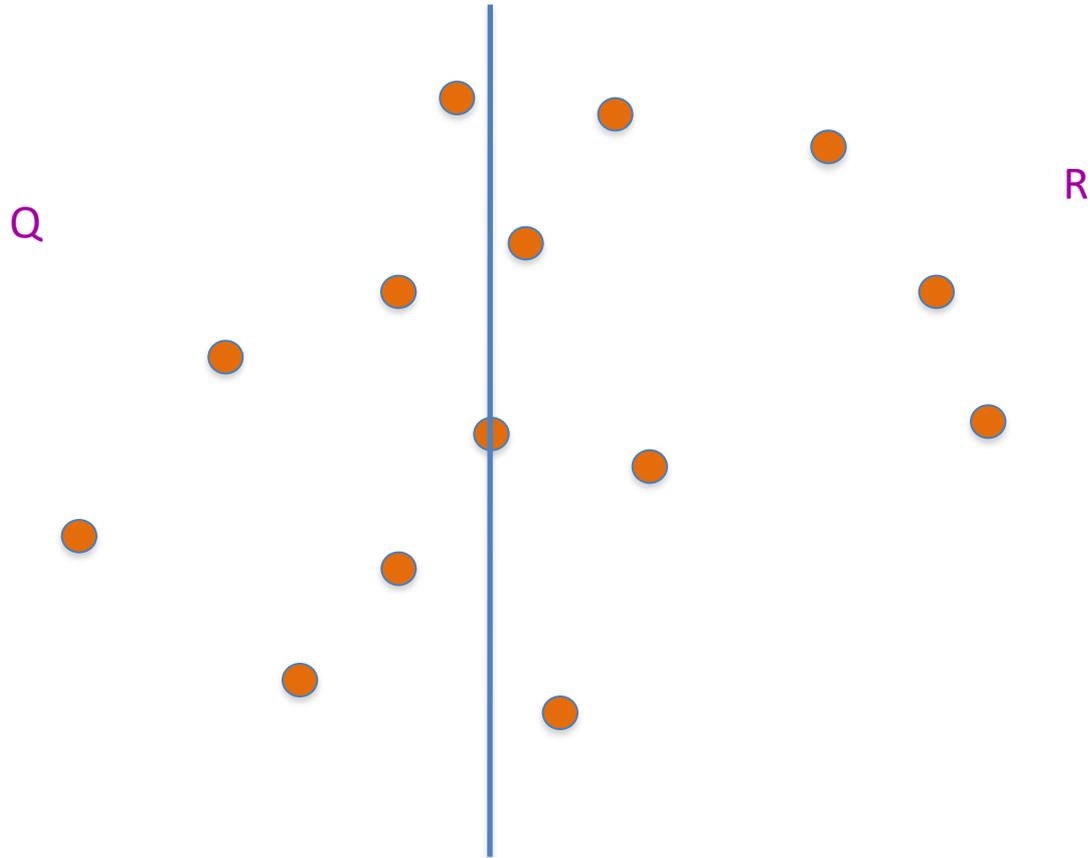
Input: n 2-D points $P = \{p_1, \dots, p_n\}$; $p_i = (x_i, y_i)$

$$d(p_i, p_j) = ((x_i - x_j)^2 + (y_i - y_j)^2)^{1/2}$$

Output: Points p and q that are closest

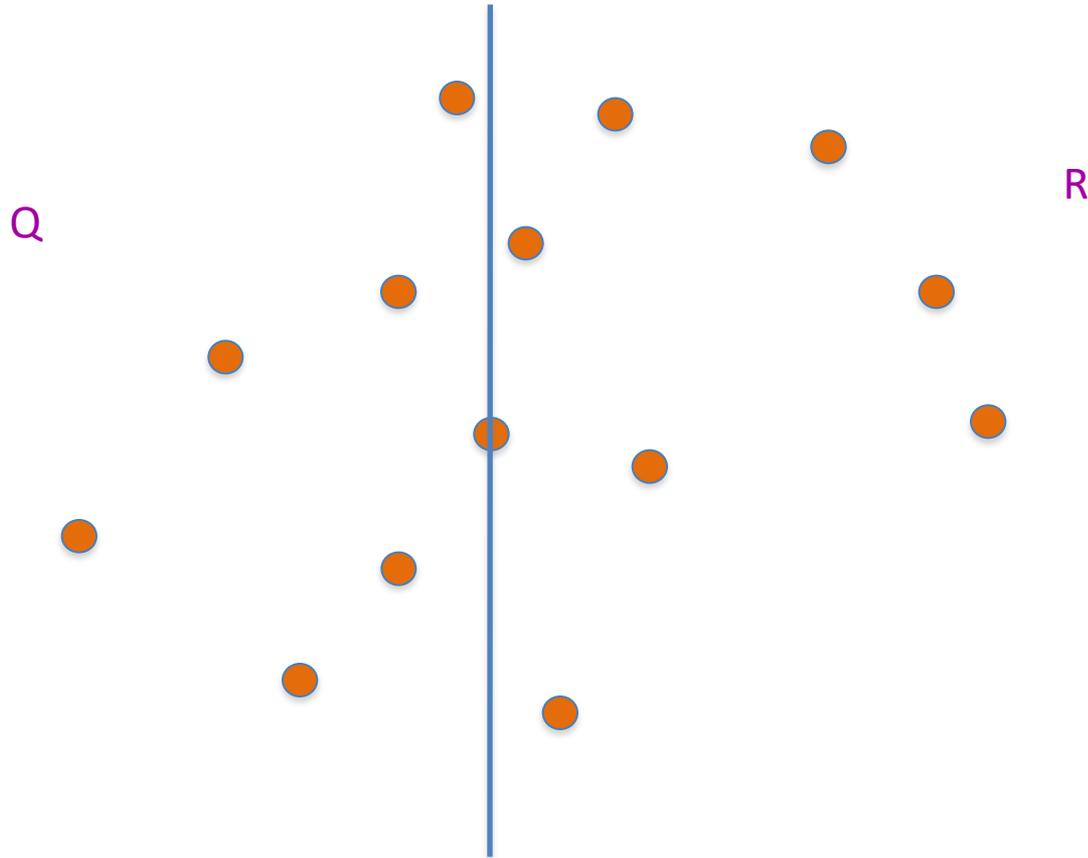


Dividing up P



First $n/2$ points according to the x -coord

Recursively find closest pairs



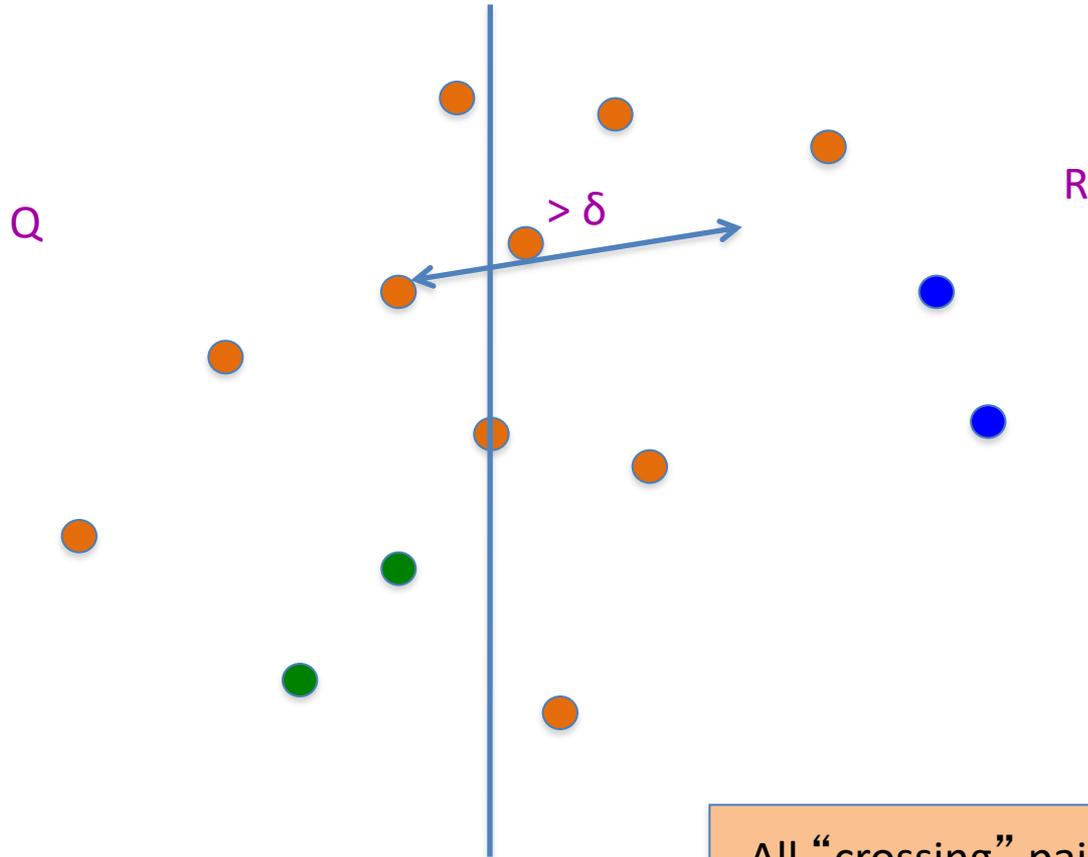
$$\delta = \min(\text{blue}, \text{green})$$

An aside: maintain sorted lists

P_x and P_y are P sorted by x -coord and y -coord

Q_x, Q_y, R_x, R_y can be computed from P_x and P_y in $O(n)$ time

An easy case

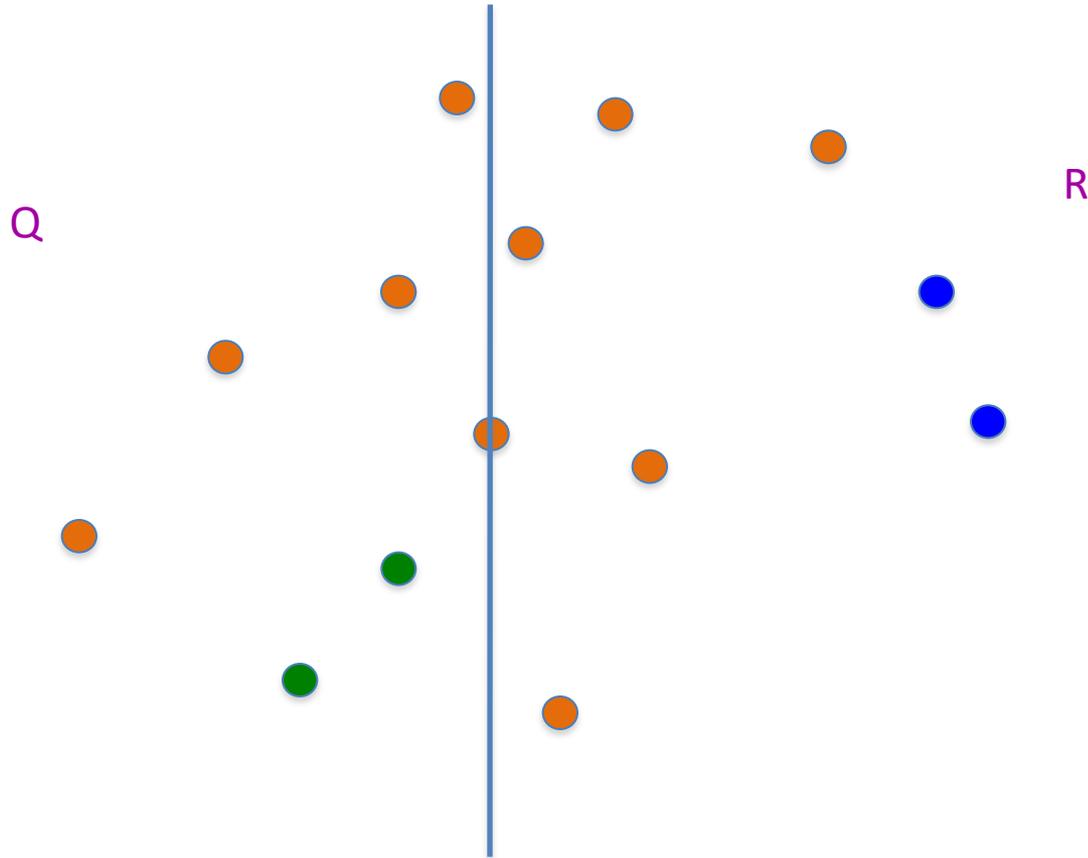


All “crossing” pairs have distance $> \delta$

$\delta = \min(\text{blue, green})$

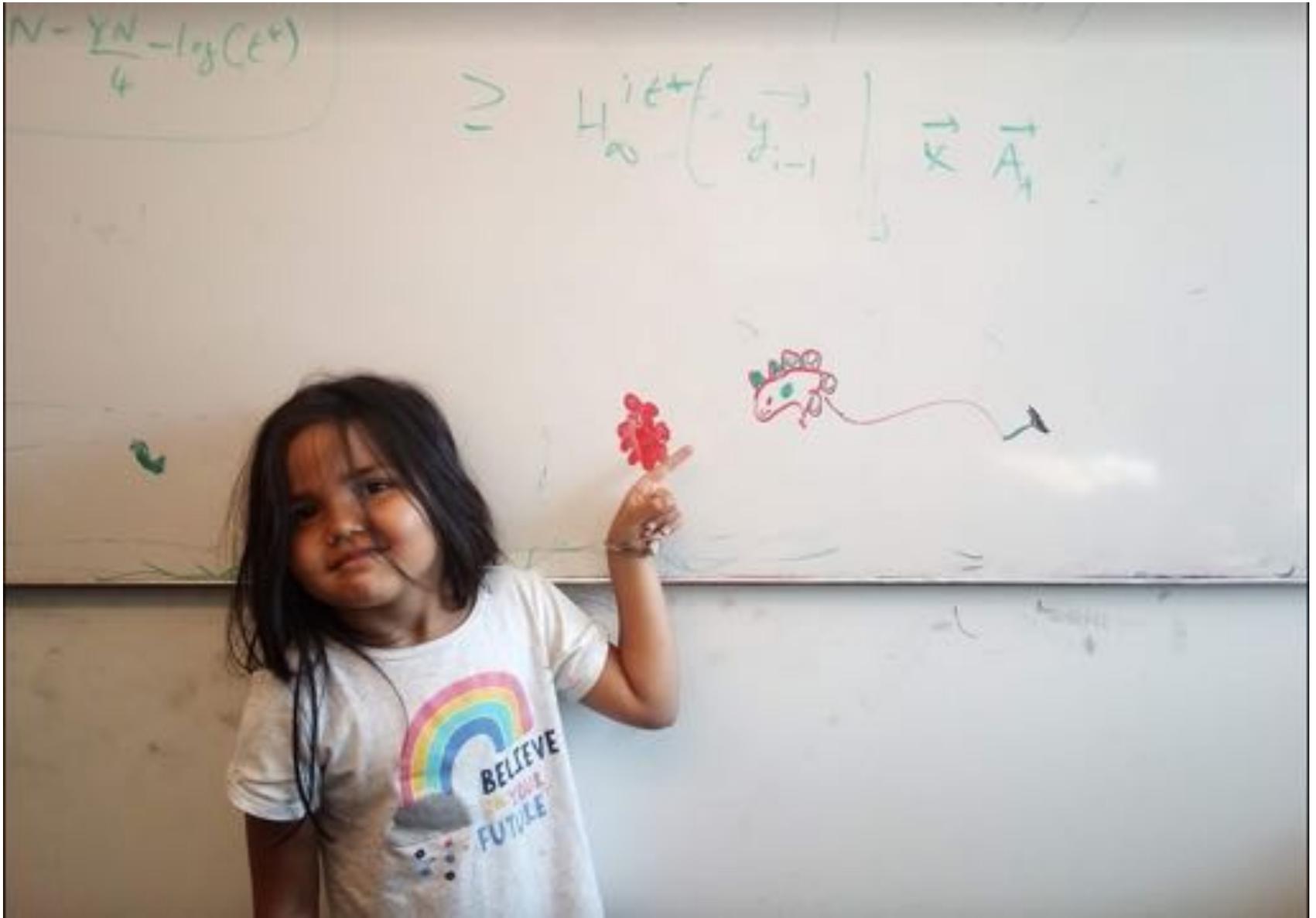


Life is not so easy though



$$\delta = \min(\text{blue}, \text{green})$$

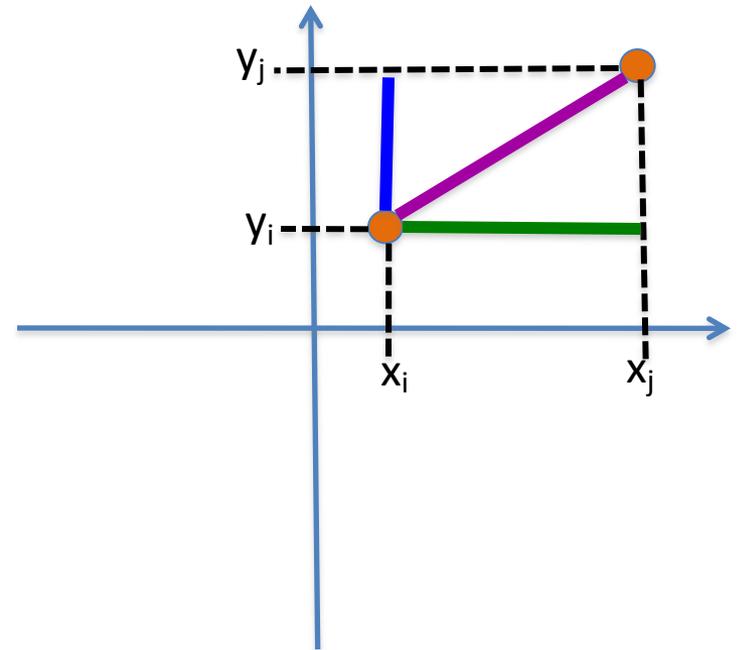
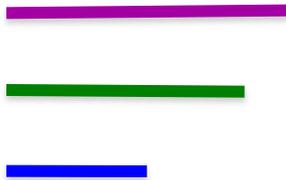
Questions/Comments?



Euclid to the rescue (?)

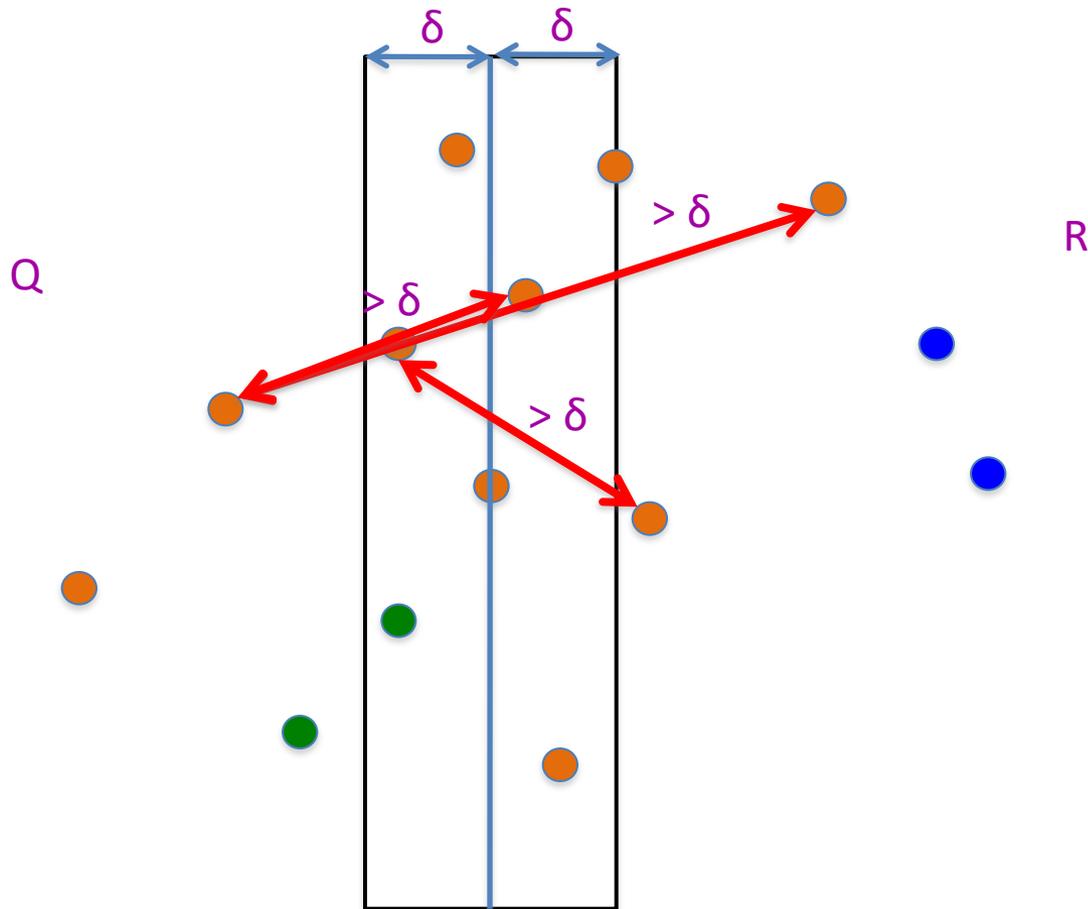


$$d(p_i, p_j) = ((x_i - x_j)^2 + (y_i - y_j)^2)^{1/2}$$



The **distance** is larger than the **x** or **y**-coord difference

Life is not so easy though



$$\delta = \min(\text{blue}, \text{green})$$

All we have to do now

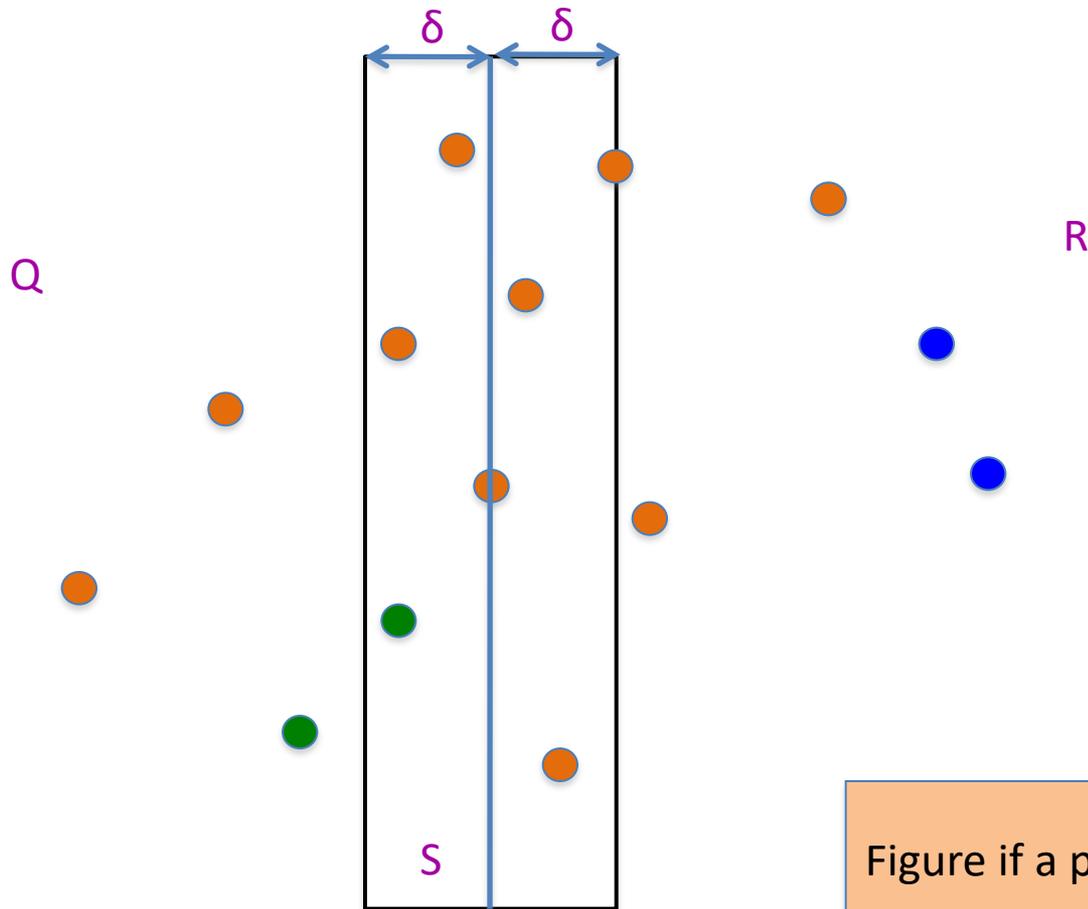


Figure if a pair in S has distance $< \delta$

$$\delta = \min(\text{blue}, \text{green})$$

The algorithm so far...

Input: n 2-D points $P = \{p_1, \dots, p_n\}$; $p_i = (x_i, y_i)$

$O(n \log n) + T(n)$

Sort P to get P_x and P_y

Closest-Pair (P_x, P_y)

$O(n \log n)$

$T(< 4) = c$

If $n < 4$ then find closest point by brute-force

Q is first half of P_x and R is the rest

$O(n)$

$T(n) = 2T(n/2) + cn$

Compute Q_x, Q_y, R_x and R_y

$O(n)$

$(q_0, q_1) = \text{Closest-Pair}(Q_x, Q_y)$

$(r_0, r_1) = \text{Closest-Pair}(R_x, R_y)$

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

$O(n)$

$S = \text{points } (x, y) \text{ in } P \text{ s.t. } |x - x^*| < \delta$

$O(n)$

return **Closest-in-box** ($S, (q_0, q_1), (r_0, r_1)$)

Assume can be done in $O(n)$

$O(n \log n)$ overall

Rest of today's agenda

Implement Closest-in-box in $O(n)$ time