Goals:

- Describe the closest points algorithm
- Argue (informally) correctness of this algorithm

Q's: Feedback on rough draft? What can you revise? Points for psets, **quizzes**, programming assignments? Weekly time?

Mon
② PSET ✓
50%

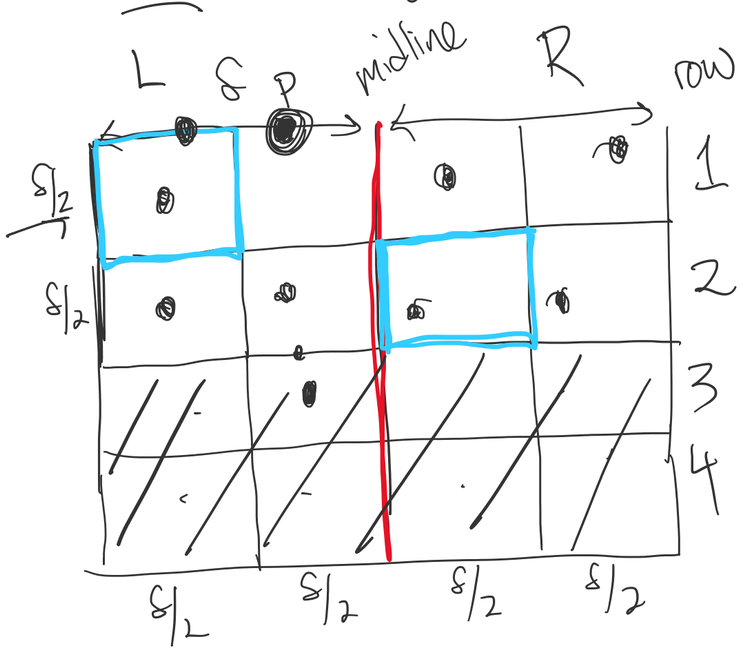
Wed
③ Self-Assess.
5-10%

7
20%

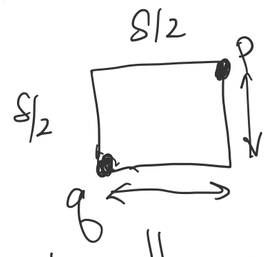
Friday
① Rough Draft ✓
20%
④ Quiz ✓ 5-10%

Lemma: Only need to look at next 7 pts from p.

Pf: *Imagine* dividing region into $\delta/2 \times \delta/2$ squares, starting at p. Each of these squares can contain at most 1 pt.



For contradiction, suppose 2 pts are in a square



$$d(p,q) \le \sqrt{\left(\frac{\delta}{2}\right)^2 + \left(\frac{\delta}{2}\right)^2} = \frac{\delta}{\sqrt{2}} < \delta.$$

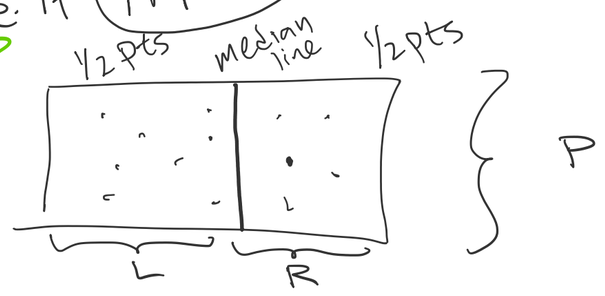
Not allowed b/c both pt are in L or both in R, so must have distance *at least* δ.

Pts in rows 3, 4 are more than δ from p. Only (7) other pts in rows 1, 2. These pts will be subsequent to p in Y_S.

ClosestPoints' Distance
ClosPt (divide + conquer)

Base case: If $|P| \le 3$ do brute force ✡

Divide:



Solve:

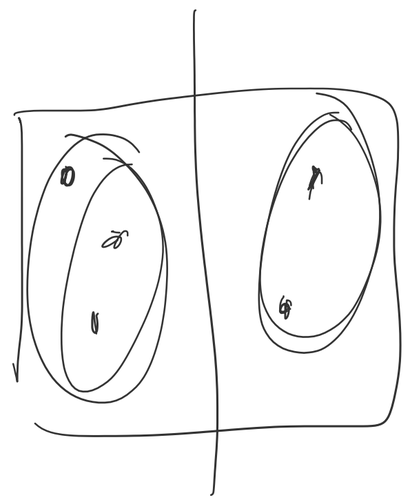
$\delta_L \leftarrow$ ClosPt(L)
$\delta_R \leftarrow$ ClosPt(R)
$\delta \leftarrow \min\{\delta_L, \delta_R\}$

Combine

$Y_S \leftarrow$ y-sorted list of pts in P with x-coord within δ of midline ✡

For $p \in Y_S$:
  Check distance from p to next 7 pts in Y_S ✡
  If distance less than δ found, set δ to new dist.

Return δ



Group: · explain in your own words why alg. works
· collect questions