CS 314 - Divide + Conquer

- Oral grading tomorrow
- New homework out tomorrow or Friday
- Comments on HW1:

Proof of correctness,
Closest Pair of Points

Let $P$ be a set of points in $\mathbb{R}^2$. Let $P = \{p_1, \ldots, p_n\}$, and $p_i = (x_i, y_i)$.

$\lVert p_i p_j \rVert = \sqrt{(x_i-x_j)^2 + (y_i-y_j)^2}$

$O(n^2)$ time

Q: What is the closest pair of points?

$O(n^2)$ $\leftarrow$ naive
Divide + Conquer Approach
Designing an algorithm

- Assume no 2 points have same x- or y-coord.
- Start by having 2 lists for points P:
  - $P_x$ - points sorted by x-coordinate
  - $P_y$ - points sorted by y-coordinate

(make sure these have indices of position in other list, so we can keep track easily)
So: Set up recursion

\( O(n \log n) \): At beginning, sort \( P \) twice to get \( P_x \) and \( P_y \)

- Find dividing line \( l \): how? look in \( P_x \) \([n/2]\)

\( O(n) \): Compute \( R \) and \( L \) - points to left and right of dividing line

\( O(n) \): Compute \( R_x \) and \( R_y \), \( L_x \) and \( L_y \) - already ordered in \( P_y \), so \( O(n) \)

- Recursively compute closest distance in \( JR \) and \( J_L \)

- Now have \( S_R \) and \( S_L \)
Prop: Let $\delta = \min \{\delta_R, \delta_L\}$. If there is $p \in \mathbf{R}$ and $q \in \mathbf{L}$ with $d(p, q) < \delta$, then $p$ and $q$ must be within $\frac{\delta}{2}$ of our dividing line $l$.

Proof: Suppose $p = (p_x, p_y)$ and $q = (q_x, q_y)$ exist. Let our line $l$ be $l = x^*$. We know $q_x - p_x < \delta$ and $p_x \leq x^* \leq q_x$. Thus $\delta > q_x - p_x \geq p_x - x^*$.

$\Rightarrow \delta > p_x - x^*$, so $p_x$ is within $\frac{\delta}{2}$ of $l$. \[ \square \]
So - only need to search a "narrow" band around $z$.

Does this always help? No: Yes

Yes: No
One more idea:

Let $S$ be points of $P$ near $l$, sorted by $y$-coordinate.

$O(n)$

Key Lemma: If $s, s' \in S$ are within 8 of each other, then $s + s'$ are within 15 positions of each other in $S$.

Why??
Proof:

Partition into boxes of size $\frac{8}{2}$

Lines

Suppose we have 2 points inside a box. How far apart are they?

$\frac{8}{2} \geq \frac{8}{2} < 8$

But these 2 points we from same side of $d$ at 8 was min distance!
So at most 1 point per box!

Now consider 2 points more than 15 positions away in $S$.

Nearest that point could be is 3 "boxes" down.

3 empty rows mean that point is at least $\frac{38}{2} = 19$ away.

So it can't be closer than 19.
Our algorithm:
After constructing $S$, compute distance between every $s \in S$ and the next 15 elements.

Key Lemma: These are only possible candidates for getting a distance $\leq \delta$

How long does this take?

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

$T(n) = 2T\left(\frac{n}{2}\right) + O(n) \implies T(n) = O(n \log n)$
Algorithm (sketch) - see p. 230 for pseudocode
- Initial sort + divide P in half
- Two recursive calls (but don't need to sort)
- Compute S & compare each pt to 15 points after it in S
Proof of Correctness:

Induction on n:
Base case - clear
For n points, our IH says \( d_k \) and \( d_x \) are closest distance for points only in \( R \) or \( L \), resp.

By our key lemma we look at all possible points per \( R \) and \( G \) with
\[
d(p, q) < \min(d_k, d_x).
\]

Since closest pair of points are either both in \( L \), both in \( R \), or one on each side, we are done - either our recursive calls found distance correctly, or our scan of \( S \) found it. \( \square \)