Preprocess (P)

Sort P by x-coord

Closest (P) // P[1...n]: list of n points

// Find distance blw closest pair in P
if (|P| ≤ 3) return min distance blw each pair
else

P_L, P_R = Partition (P)

d_L = Closest (P_L)
d_R = Closest (P_R)

d = min (d_L, d_R)

d_c = Closest Crossing (P_L, P_R)

What to eventually return?

return min (d_L, d_c, d_c)

(1) Partitioning:

How to partition into P_L, P_R?

Partition (P)

Sort P by x-coord

P_L = P[1...n/2]

P_R = P[n/2+1,...,n]

return P_L, P_R

Problem? Partition() called for every recursive step.
So sort required each time.

Solution? Sort \( P \) once at the beginning!

/Preprocess (P) /\add to original code
Sort \( P \) by x-coord  remove from partition

(2) What if subset contains only one point?
(3) ClosestCrossing

\[ \{ \quad \} \]

Find closest distance b/w pairs that cross partition

\[ \{ \quad \} \]

\[ P_L \quad P_r \]

Simplest approach?

\[ \Rightarrow \text{compare all } \frac{n}{2} \text{ points in } P_L \text{ with all } \frac{n}{2} \quad P_r \]

How many comparisons? \( \left( \frac{n}{2} \right) \left( \frac{n}{2} \right) = \frac{n^2}{4} \)

Necessary?

\[ \text{No! Again, probably not} \]

\[ \text{necessary to compare} \]

\[ \text{these 2} \]

So which pairs of points should we compare?

Hint: Look back at ClosestCrossing()

We know closest distance on left \( d_L \)

" " " " right \( d_R \)

So we know \( \min(d_L, d_R) \) is closest distance of non-crossing points!
(Partitioning w/ $\left( \frac{n}{2}, \frac{n}{2} \right)$ or $\left( \frac{n+1}{2}, \frac{n}{2} \right)$ or $\left( \frac{n}{2}, \frac{n}{2}+1 \right)$)

Suppose subset has 5 points:

1. $\left[ \begin{array}{c} 0 \end{array} \right]$ 2. $\left[ \begin{array}{c} 1 \end{array} \right]$ 3. $\left[ \begin{array}{c} 2 \end{array} \right]$ 4. $\left[ \begin{array}{c} 3 \end{array} \right]$ 5. $\left[ \begin{array}{c} 4 \end{array} \right]$ - compute directly.

| 2nd 1st | Part | Part |

Can't compute distance for single points.

So, set base case at $n=3$.

Another solution:

Have 2 base cases:

- if $n=1$ return $\infty$
- if $n=2$ return $\text{dist}(p_1, p_2)$

$\Rightarrow$ change code!!
So care only about crossing pairs that are less than \( \min(d_l, d_r) \).

Add to code: \[ d = \min(d_l, d_r) \]

**LIST**

**Observations for closestCrossing()**:  
- **Compare only points**:
  1. within horizontal distance \( d \) from the partition
  2. vertical from each other!

**Example**: Suppose \( d_p = 20 \), \( d_l = 10 \) \( \Rightarrow d = 10 \)

Note: Could also consider only points within distance \( d \) from each other (instead of from partition), but slightly more complex to find these.
How to find points within d from partition?
⇒ Points are already sorted by x-coord, so just check if x-coord is within d of x-coord of partition.

\[ \left( x_{1}, y_{1} \right) \cdots \left( x_{\frac{n}{2}}, y_{\frac{n}{2}} \right) \left( x_{\frac{n}{2} + 1}, y_{\frac{n}{2} + 1} \right) \cdots \left( x_{n}, y_{n} \right) \]

\[ \lceil \frac{x_{\frac{n}{2}} + x_{\frac{n}{2} + 1}}{2} \rceil \]

Now, how many comparisons?

Worst-case:

ex: All points clustered around partition:

\[ p \]

Notice: Distance b/w pairs within \( p_{l}, p_{r} \) > distance b/w crossing pairs.

\[ \frac{n}{2} \cdot \frac{n}{2} \]

At most \( \left( \frac{n}{2} \right) \cdot \left( \frac{n}{2} \right) \) comparisons!

⇒ Should compare \( p \) + \( a \), but not \( p \) + \( b \)
⇒ So also consider vertical distance!

\[ \{ \} \]

If vertical distance > d, don't compare!
Observation #2:

Add to list of observations:
- within vertical distance $d$ from each other.

How many comparisons?
" such pairs of points?

Consider one point $p$:

ex: partition at $18$, $d=10$, $p=(15,80)$

$8-d-18-d-25$

Need only consider points within $p$'s box.

How many such points?

Split $p$'s box into 16 regions:

At most how many other points in $p$'s box?
Max per region?

Note: every pair must be at least \( \geq \frac{d}{2} \) apart.

Let's try to put 2 in one region. Farthest we can place 2 is at corners.

\[
\begin{align*}
&\left[ \begin{array}{c}
\frac{d}{2} \\
\frac{d}{2}
\end{array} \right] \\
\end{align*}
\]

Are they \( \geq d \) apart?

\[
\text{dist} = \sqrt{\left(\frac{d}{2}\right)^2 + \left(\frac{d}{2}\right)^2} = \sqrt{\frac{2d^2}{4}} = \frac{d}{2} > \frac{d}{2} < d
\]

At most 1 point in each region.

15 points to compare to \( p \).

Total \( \# \) comparisons in \text{ClosestCrossing}(\).

For every point in \( P \) that is within horizontal distance \( d \) from partition, \( \frac{n}{2} \)

\[
\Rightarrow \frac{n}{2} \times 15 \leq \frac{n}{8}
\]

Much better than \( \frac{n^2}{4} \).