# Lecture 21

## COL 351: Analysis and Design of Algorithms

**Lecture 21**

## Closest Pair of Points

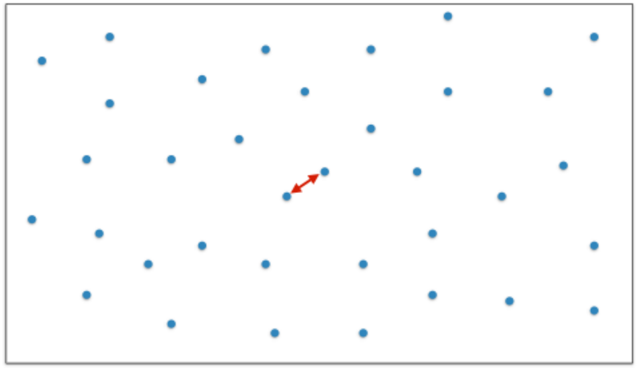
# (Divide and Conquer strategy)

## Closest Pair of Points (or Minimum pairwise distance)

**Given:** A set $P$ of $n$ points in x-y plane.

**Output:** A pair of points in $P$ at minimum distance, or $\min_{a \neq b \in P} distance(a, b)$.

Example:



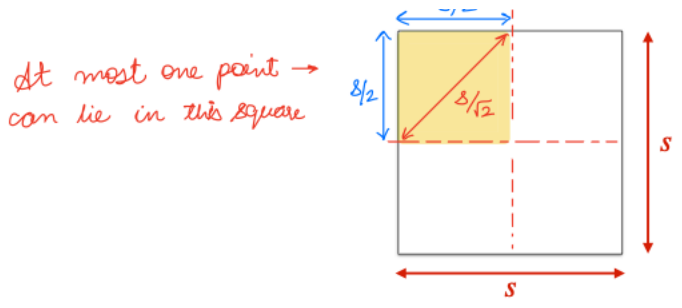
Trivial : $O(n^2)$ time

## Subproblem

**Given:** A set $P$ of points in x-y plane satisfying $\min_{\substack{(a,b) \in P \times P \\ a \neq b}} distance(a, b) \geq s$

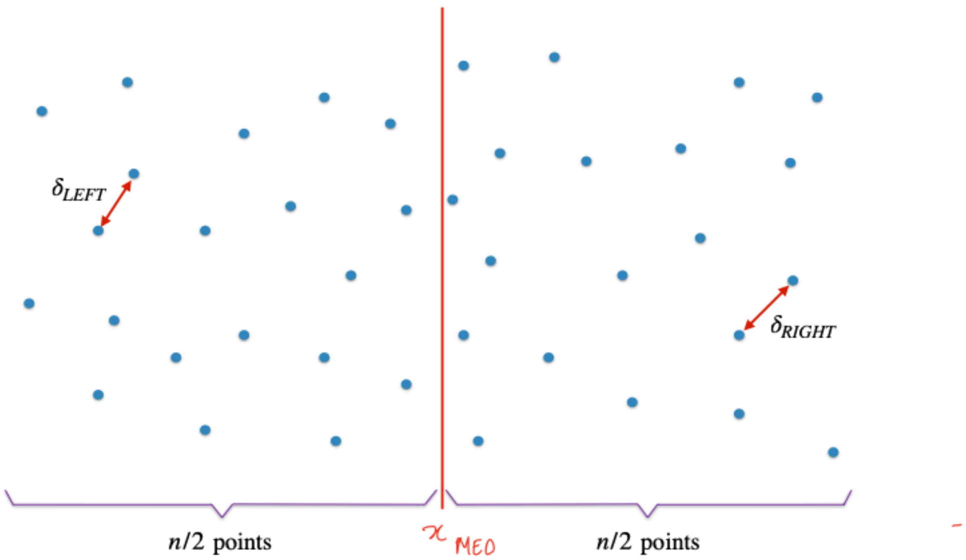
**Question:** What is the maximum number of points that can fit in a square of size $s \times s$ ? "FOUR"

$s/2$



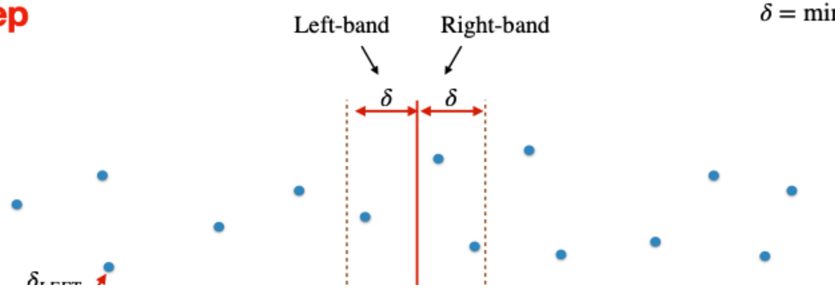

## Divide and Conquer (Divide step)

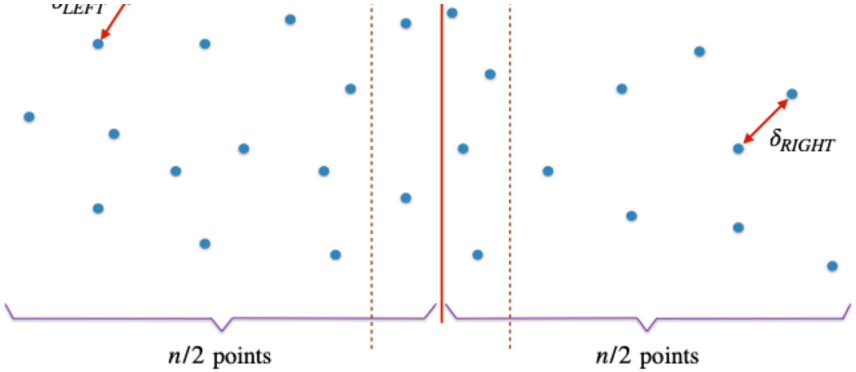
$\delta = \min(\delta_{LEFT}, \delta_{RIGHT})$




## Combine Step

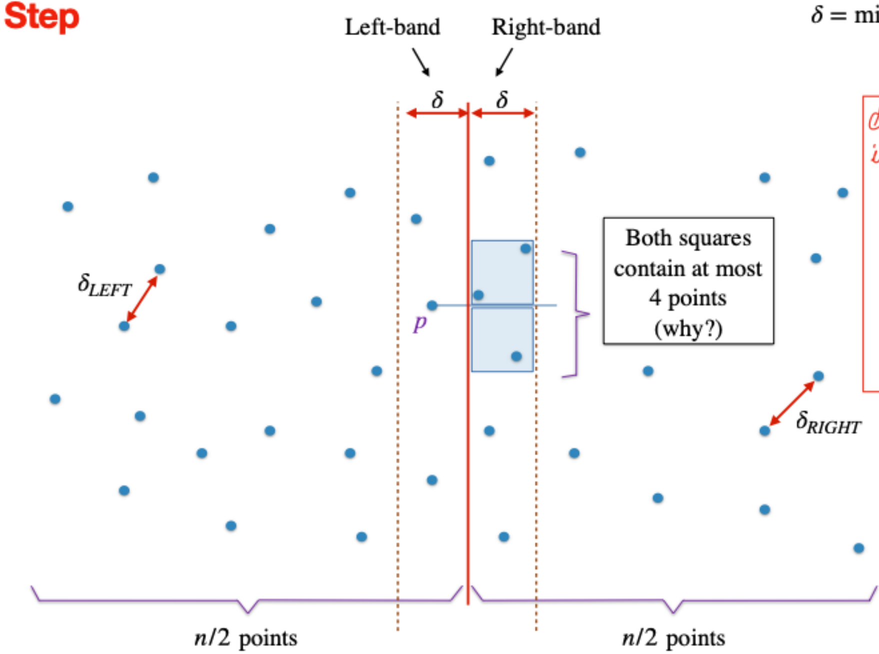
$\delta = \min(\delta_{LEFT}, \delta_{RIGHT})$






## Combine Step

$\delta = \min(\delta_{LEFT}, \delta_{RIGHT})$




All points $(x,y)$ in Right Band s.t. $|y - y(p)| \leq \delta$ are contained in the two highlighted squares.

MinPairwiseDistance($P$)

1. **If** ($|P| = 1$) return $\infty$
2. $x_{MED}$ = Median of points in $P$ according to x-coordinate } $O(n)$
3. ($P_{LEFT}$, $P_{RIGHT}$) = Partition of $P$ by $x_{MED}$ } $O(n)$
4. $\delta_{LEFT}$ = MinPairwiseDistance($P_{LEFT}$) } $2T(n/2)$
5. $\delta_{RIGHT}$ = MinPairwiseDistance($P_{RIGHT}$)
6. $\delta = \min(\delta_{LEFT}, \delta_{RIGHT})$

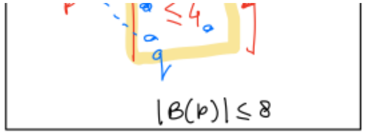
$T(n) = 2T(\frac{n}{2}) + n\log n$

7. Left-band = $\delta$-length band of $P_{LEFT}$ } $O(n)$
8. Right-band = $\delta$-length band of $P_{RIGHT}$

9. $B$ = Points in Right-band sorted according to y-coordinate } $O(n \log n)$

10. **ForEach**( $p \in$ Left-band ):
    $B(p)$ = Points in $B$ whose y-coordinate differ from that of $p$ by at most $\delta$ ← Binary search over "B"
    $\delta_p$ = minimum distance b/w $p$ and points in $B(p)$
    **If** $(\delta_p < \delta)$: $\delta = \delta_p$

11. Return $\delta$




**Algorithm**

$$T(n) = 2T(n/2) + \cancel{O(n \log n)} \; cn \log n$$

---

$\boxed{\log \frac{n}{2} \leq \log n}$

$$T(n) = 2\left(2T\left(\frac{n}{4}\right) + c\frac{n}{2}\log\frac{n}{2}\right) + cn\log n$$

$$\leq 4T\left(\frac{n}{4}\right) + 2cn\log n$$

$$\leq 8T\left(\frac{n}{8}\right) + 3cn\log n$$

$$\vdots$$

bcoz $2^i = n$

$i = \log_2 n$

$$\leq 2^i\, T\left(\frac{n}{2^i}\right) + i\,cn\log n$$

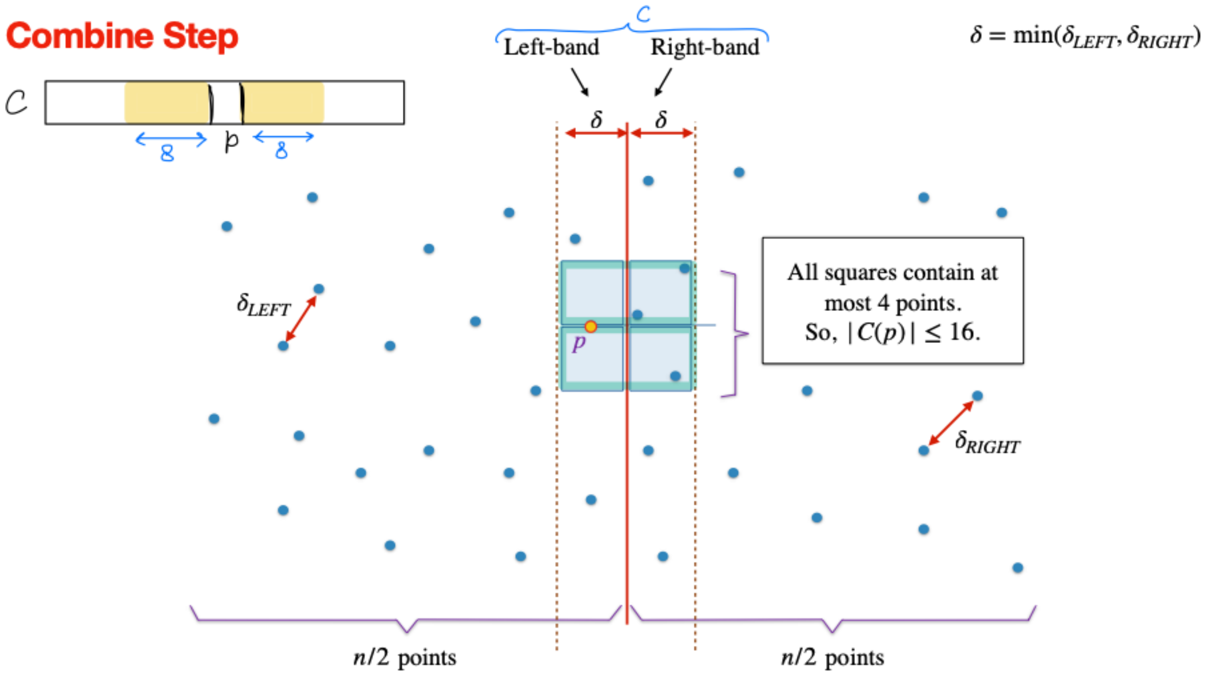
$$\leq n\,T(1) + cn\log^2 n = O(n\log^2 n)$$

## Minimum pairwise distance

**Result:** Given a set $P$ of $n$ points in x-y plane, we can compute $\min_{a \neq b \in P} distance(a, b)$ in $O(n \log^2 n)$ time.

**Question:** Can we improve the bound to $O(n \log n)$ time?

## Combine Step




MinPairwiseDistance($P$)

**If** ($|P| = 1$ ) return $\infty$
$x_{MED}$ = Median of points in $P$ according to x-coordinate
$(P_{LEFT}, P_{RIGHT})$ = Partition of $P$ by $x_{MED}$
$\delta_{LEFT}$ = MinPairwiseDistance($P_{LEFT}$)
$\delta_{RIGHT}$ = MinPairwiseDistance($P_{RIGHT}$)

Time complexity follows the relation

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

6. $\delta = \min(\delta_{LEFT}, \delta_{RIGHT})$
7. Left-band = $\delta$-length band of $P_{LEFT}$
8. Right-band = $\delta$-length band of $P_{RIGHT}$

9. $C$ = Points in (Left-band $\cup$ Right-band) sorted according to y-coordinate

Takes O(n) time if points are sorted during pre-processing stage

O(n)

10. **ForEach**( $p \in C$):
    $C(p)$ = Points in $C$ whose y-coordinate differ from that of $p$ by at most $\delta$.
    $\delta_p$ = minimum distance b/w $p$ and points in $C(p)$.
    **If** ($\delta_p < \delta$): $\delta = \delta_p$

11. Return $\delta$

$C(p)$ is subset of 8 predecessors and 8 successors of $p$ in '$C$'

Time to compute $C(p)$ given $C$ is $O(1)$

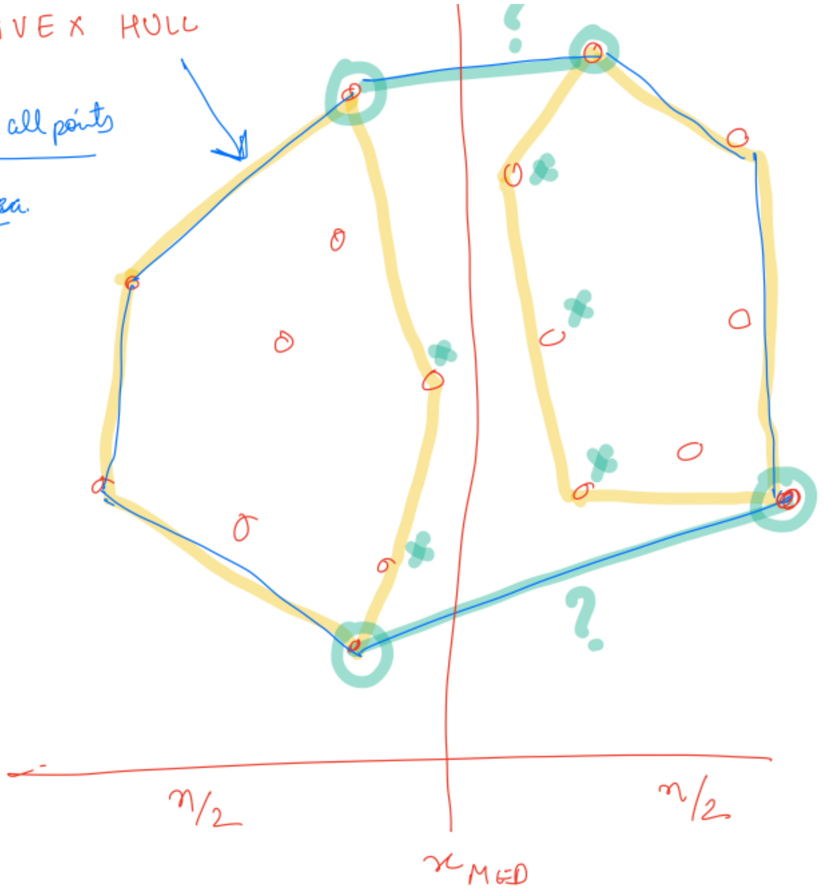
**Improved Algorithm**

## Minimum pairwise distance

**Result:** Given a set $P$ of $n$ points in x-y plane, we can compute $\min_{a \neq b \in P} distance(a, b)$ in $O(n \log n)$ time.

CONVEX HULL

- contains all points
- min area




$2T(n/2)$