10/25/18

⇒ MASTER THEOREM:

- $T(n) = \underset{a}{3}\,T\left(\underset{b}{n/4}\right) + \underbrace{n\log n}_{f(n)}$

$f(n)$ v/s $n^{\log_b a}$

$n\log n$ $\quad$ $n^{0.793}$

Case (3) ⇒ $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for $\varepsilon = 0.2$

for some $c < 1$, $a f(n/b) \le c f(n)$

$3\frac{n}{4}\log\frac{n}{4} < n\log n \rightarrow c =$

⇒ $T(n) = \Theta(n\log n)$

- $T(n) = \underset{a}{2}\,T\left(\underset{b}{n/2}\right) + \underbrace{n\log n}_{f(n)}$

$f(n)$ v/s $n^{\log_b a}$

$n\log n$ v/s n

What does polynomi
larger mean?
Say n^2 & $n^{2+\varepsilon}$ wit
then $n^{2+\varepsilon}$ is polynom
larger than n^2

Doesn't quite fit with case 3 because stricter rule need to have polynomially larger value.

Used a lot in Computational Geom...

DIVIDE AND CONQUER:
CLOSEST PATH OF POINTS

? Given n points, find the pair of points with the smallest distance between them.

→ Naive solution: go through every pair → $\Theta(n^2)$

1D:

Sort, compute distances between consecutive points recalibarating the smallest one.
↳ sort by x value
↳ closest point will be immediate neighbour (left or right)

$\Theta(n\log n)$ ⇓ $\Theta(n)$

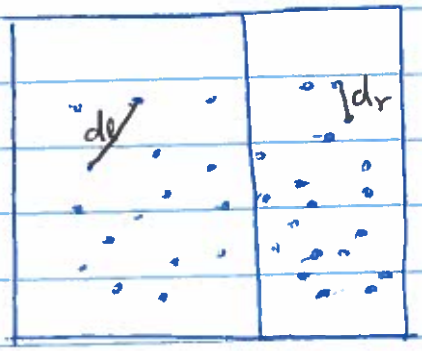
$\Theta(n\log n)$

$\Downarrow$

$\Theta(n\log n)$

- Can't be applied to 2-D right away. So we use Divide & Conquer!

→

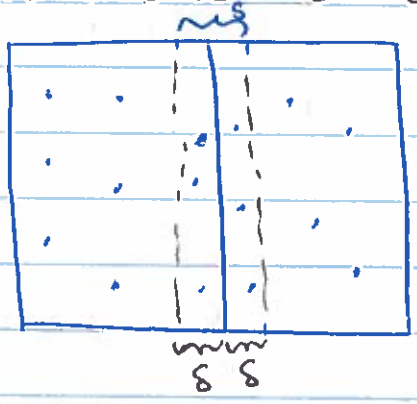


Divide space such that $\sim n/2$ points on each side

- Find closest pair in each side recursively

★ – Find closest pair with one point on each side
– Return the best of the three solutions.

? Doesn't this reduce to comparing all points?



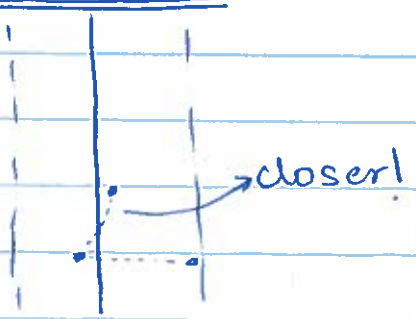
Check in a smaller region → δ regions.

$\delta = \min(\delta_L, \delta_R)$

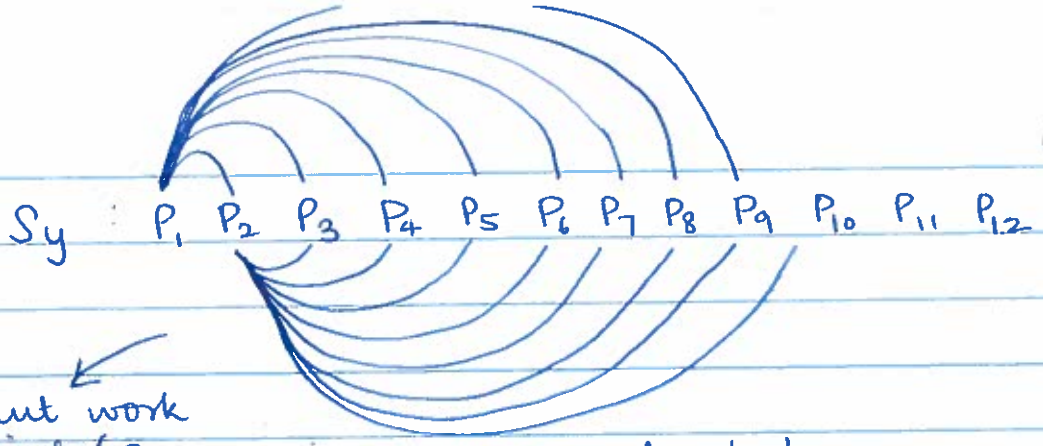
Only need to consider set S of points within $\delta = \min(\delta_L, \delta_R)$ of dividing line.

→ Can we do better?

- Sort S by y coordinate to get S_y.
Computing distances between consecutive points in S_y doesn't work



Instead, compute distances to points within 8 positions in S_y



Constant work per point (8 comparisons) → order 'n'

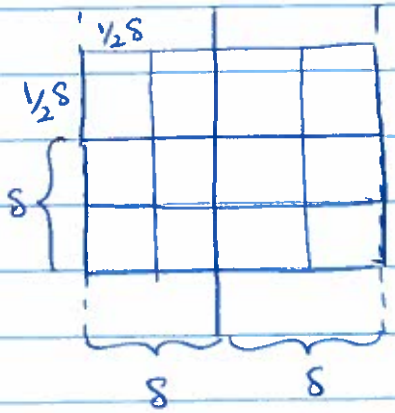
- distance = Euclidean distance

- Called Strong's Algorithm.
- On the halves it is a recursion and middle strip is this way.

→ Claim: If two points are more than 8 positions apart in Sy, then the distance between them is $> \delta$

Euclidean distance.

- Proof: Visualize $\frac{1}{2}\delta \times \frac{1}{2}\delta$ boxes near dividing line. Points in Sy must lie in these boxes.



Key Observation: There is at most one point in every box.

↓

Why so? If say  Then they would be closer than δ and our δ wouldn't be δ (contradiction)

If >8 positions apart ⟹ >2 rows apart

- Complexity ⟹ $2T(n/2) + O(n)$ ⟹ $\boxed{T(n) = O(n \log n)}$