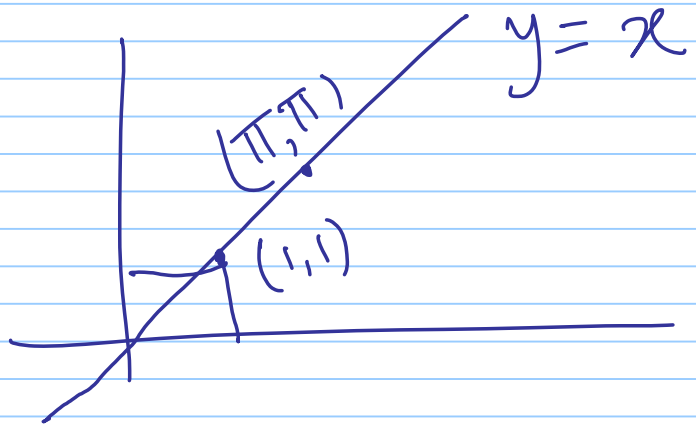
Note Title 03-06-2019

Introduction to Computational Geometry (CG)

CG – what is it?

Points, lines, line segments, circles, ...



finite precision

rational coordinates

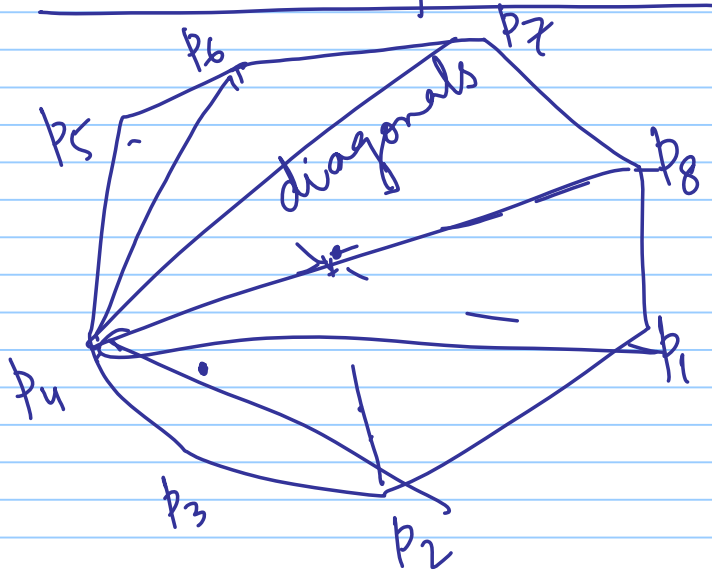
LEDA

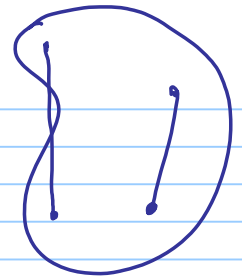
CGAL

Assumptions: (1) no three or more points are colinear

(2) no four or more points are cocircular.

Area computation:





Convex polygon on n vertices.

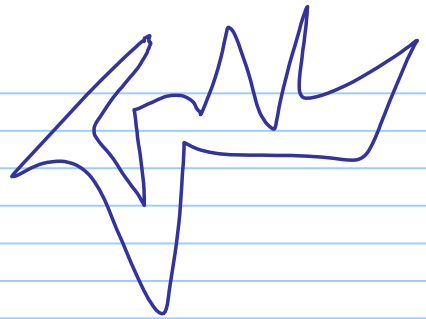
The vertices are given in a clockwise order (CW) / CCW

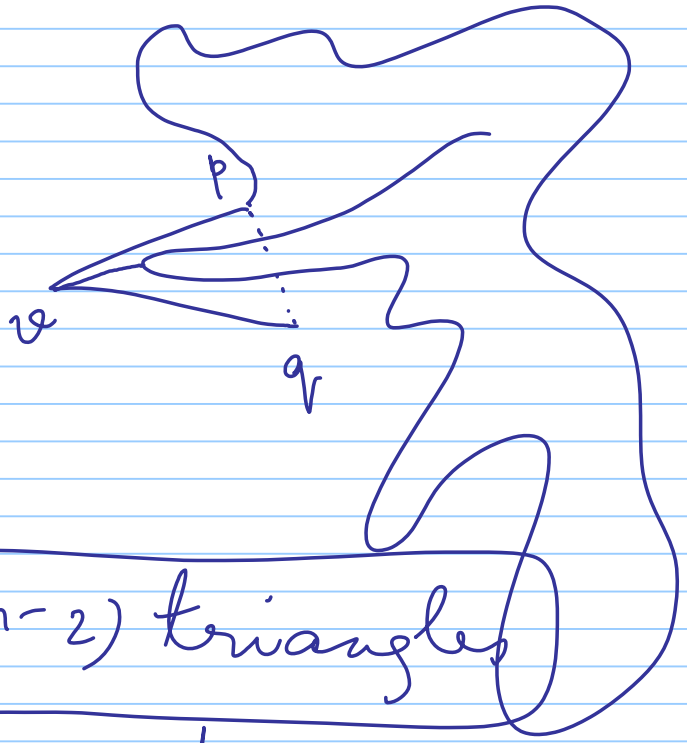
P is the input.

Goal: compute the area of P.

$(n-2)$ triangles by drawing $n-3$ non-crossing diagonals.

Simple polygon:





Ex: Show (using induction) that any simple polygon of n vertices can be divided into $\boxed{(n-2) \text{ triangles}}$ by drawing $(n-3)$ non-crossing diagonals.

Closest Pair: **Input:** a set of n points $P = \{p_1, p_2, \ldots, p_n\}$ in $\mathbb{R}^2$ with the usual assumptions.

Goal: Find the closest pair of points / minimum distance among all pairs of points.

Trivial: Look at all pairs of points.

$$\binom{n}{2} \approx O(n^2)$$

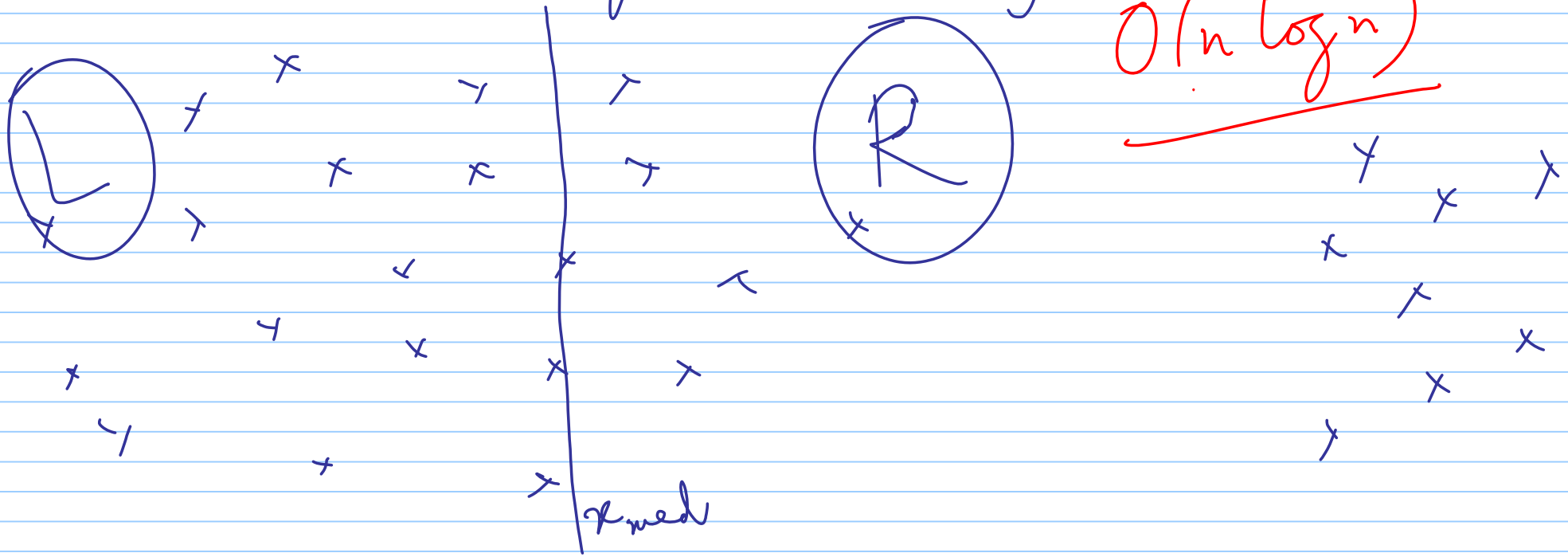
We can find median in $O(n)$:

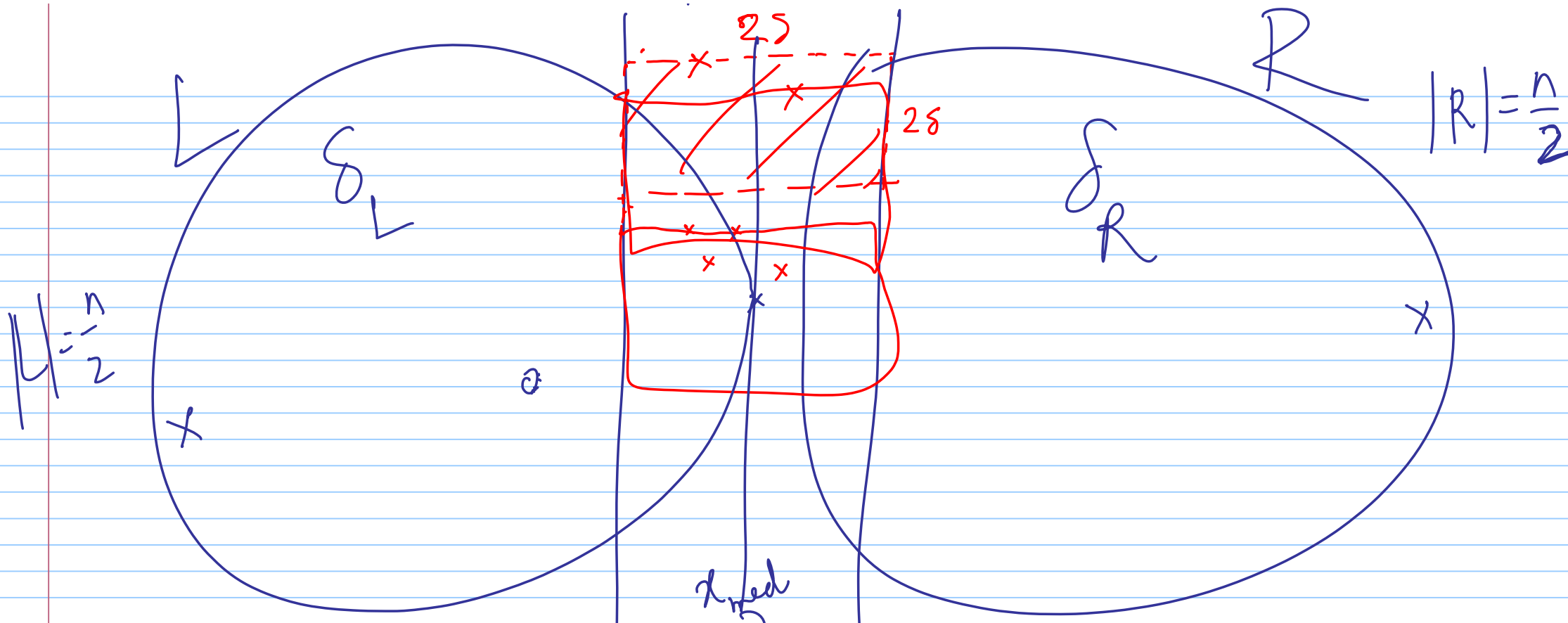
$$T(n) \le 2\boxed{T(n/2)} + O(n)$$

divide-and-conquer

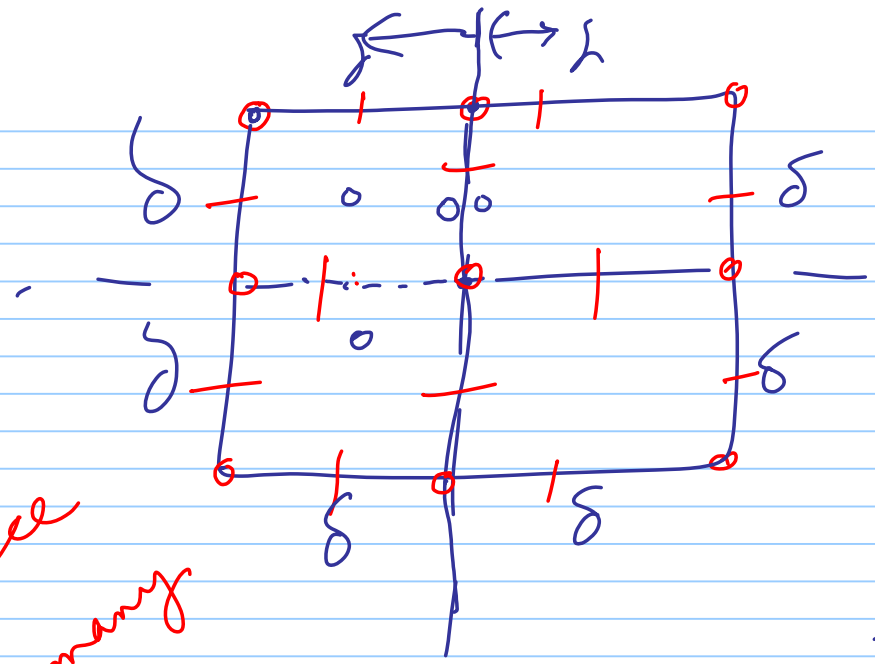
$$T(n) = O(n \log n)$$

$O(n \log n)$





$$\delta = \min\{\delta_L, \delta_R\}$$



$2\delta \times 2\delta$ square

Q: Square of $2\delta \times 2\delta$

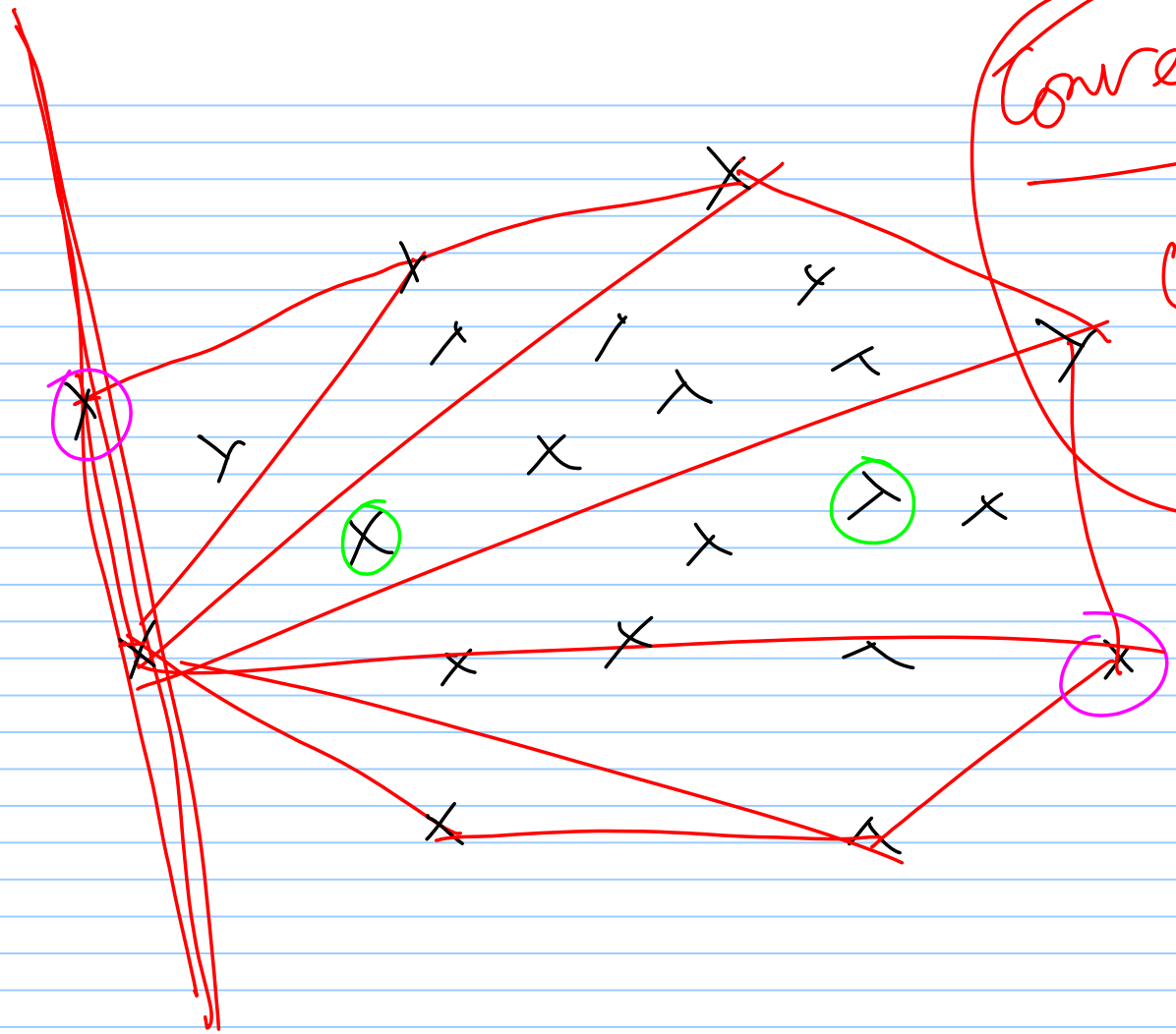
Goal: Place points inside Q s.t. you place max^{m} # points as per the rule stated.

Ex: you can place only $O(1)$ many points.

Farthest pair. $P = \{p_1, p_2, \ldots, p_n\}$

$p_i \in \mathbb{R}^2$

Goal: Find the pair of points that is farthest apart.



Convex hull

$O(n \log n)$

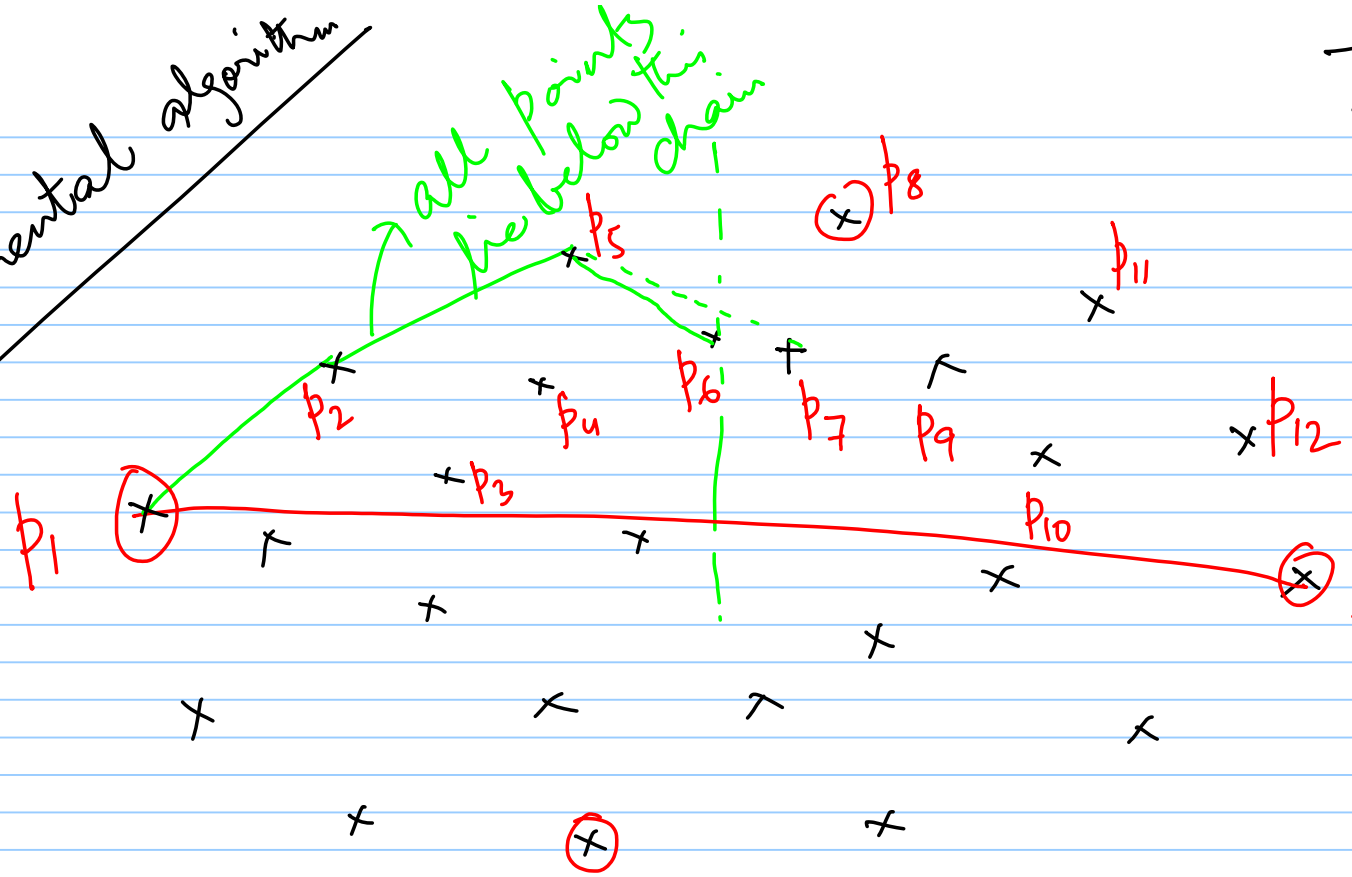
Convex hull: Input: $P = \{p_1, p_2, \ldots, p_n\}$

$p_i \in \mathbb{R}^2$

Goal: Compute $CH(P)$

Def: **Convex hull:** It is the smallest convex set that contains P.

Incremental algorithm

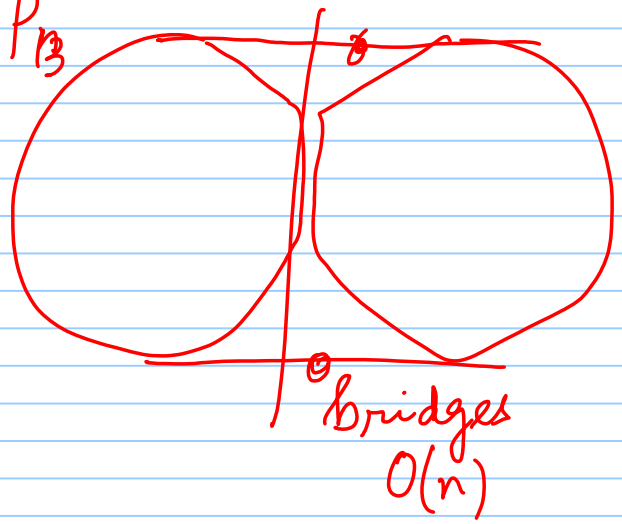


Target: $O(n \log n)$

Data Structure (?)

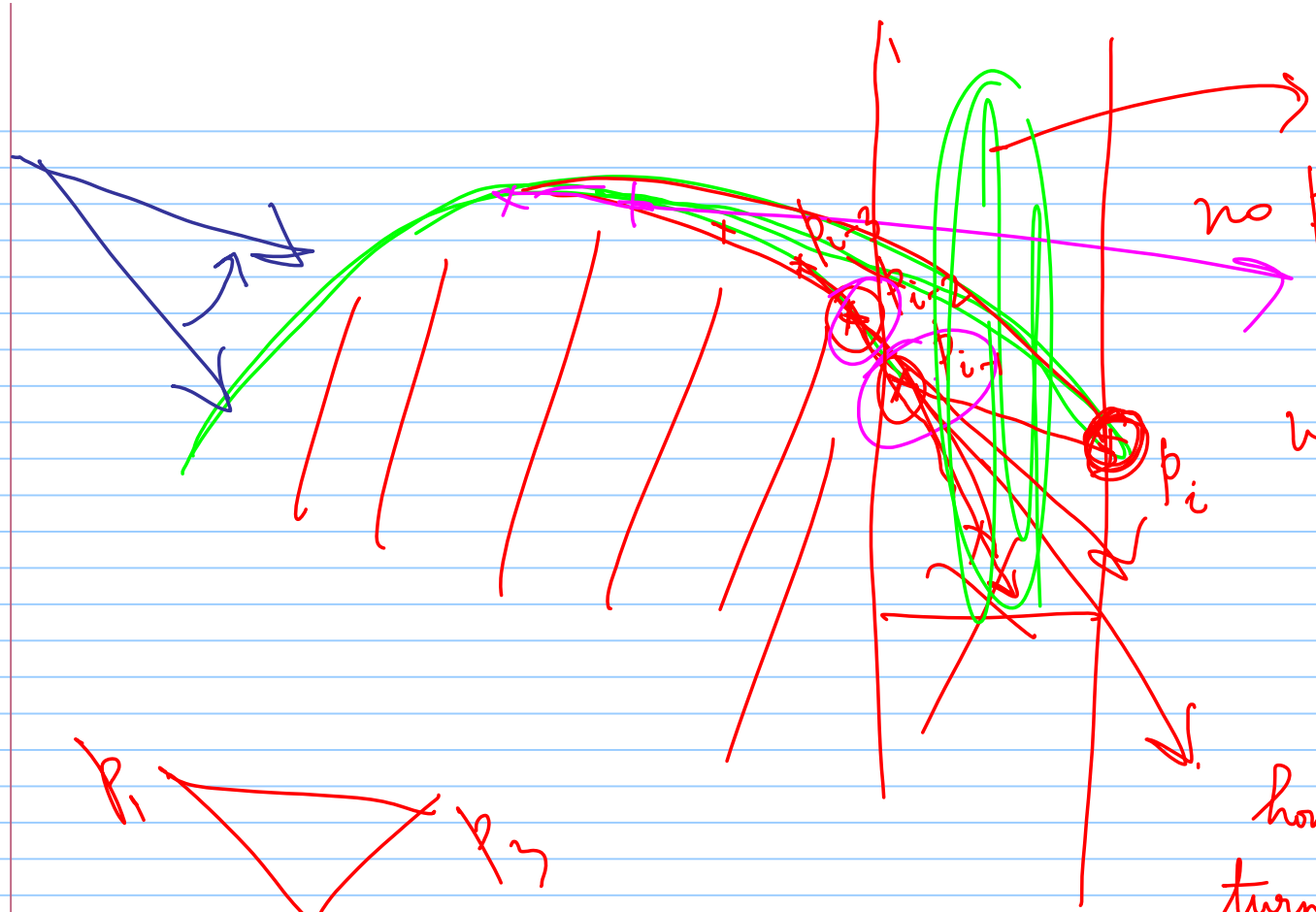
Stack

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



Algorithm:

1. Sort the points acc. to their x-coordinates. — $O(n \log n)$
2. Pick up the leftmost & rightmost points and divide the point set into two halves — one that lie below the line segment $\overline{p_1 p_n}$ and the one that lies above. $O(n)$
3. Use a stack to store the right to left order of the points on the hull
4. In a loop, pick up points from the x-sorted order, and construct the new chain till we exhaust all points on the upper half.
5. Do accordingly for the lower half.



no point in this zone

why (?)

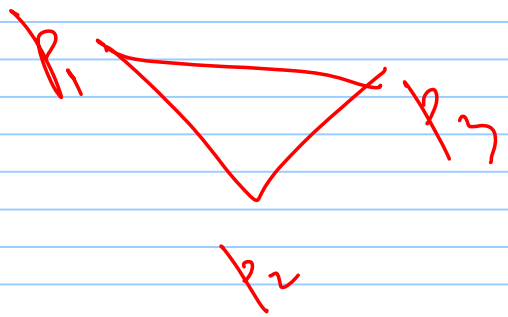
$$\begin{vmatrix} 1 & p_1^x & p_1^y \\ 1 & p_2^x & p_2^y \\ 1 & p_3^x & p_3^y \end{vmatrix}$$

$O(1)$

Geometric Primitive: Ex

Given points p_1, p_2, p_3

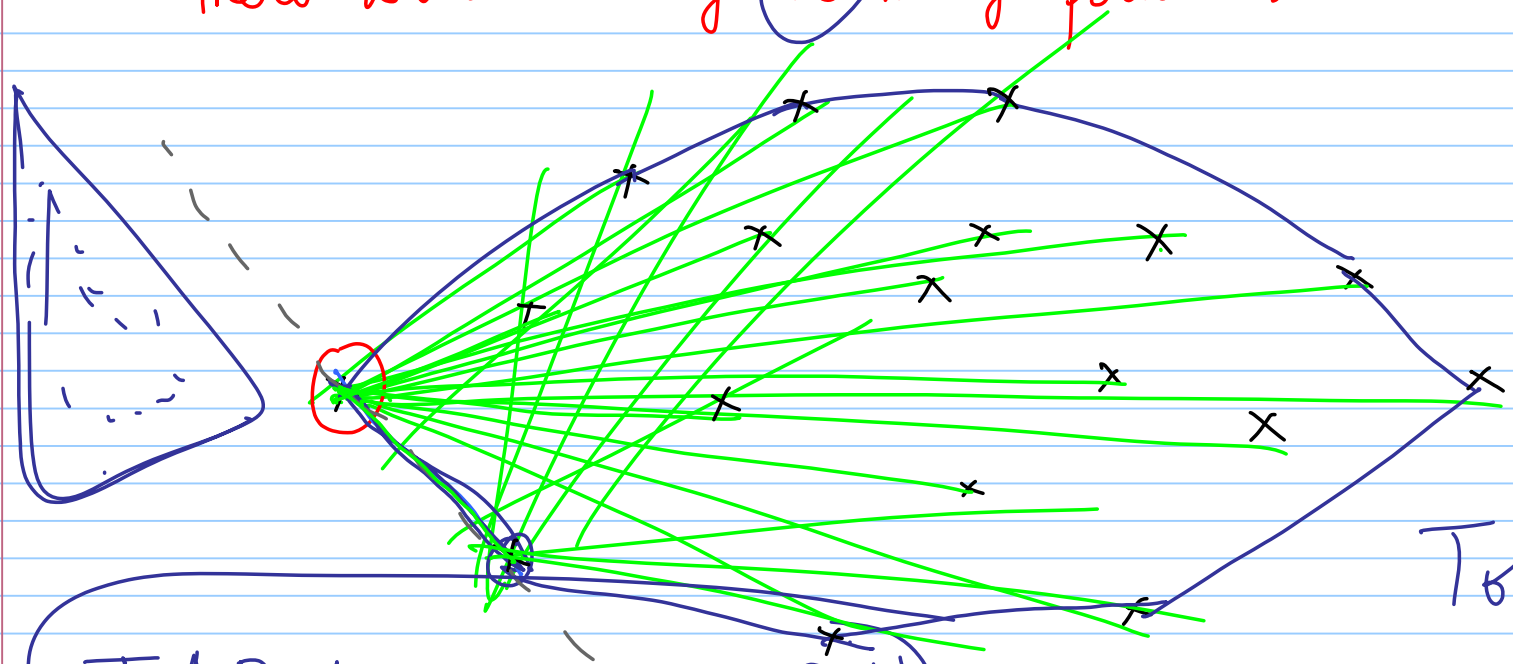
how do we figure out left & right turns from line segment $\overrightarrow{p_1 p_2}$ to p_3?



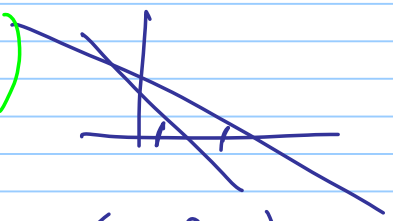
Output sensitive algorithm

- We have got an optimal algorithm for convex hull.
- There will be only h many points on the convex hull.

Gift wrapping technique.



$O(n)$



Total: $O(nh)$

JARVIS MARCH

- $O(n \log n)$ Graham's scan
- $O(nh)$ Jarvis March
- $O(?)$ $n \log h$

$n^2 = O(n^3)$

$n^2 = O(n^2 \log n)$

$n^2 = O(n^2)$

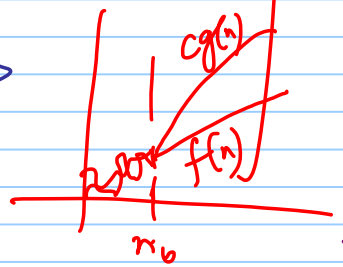
$f(n) = O(g(n))$

$f(n) \le c\, g(n)$ +ve constant

$\forall n \ge n_0$

Timothy Chan

1990s



big-Oh

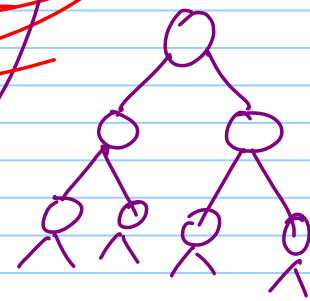
Geometric Series:

$$1 + r + r^2 + \cdots + r^n$$

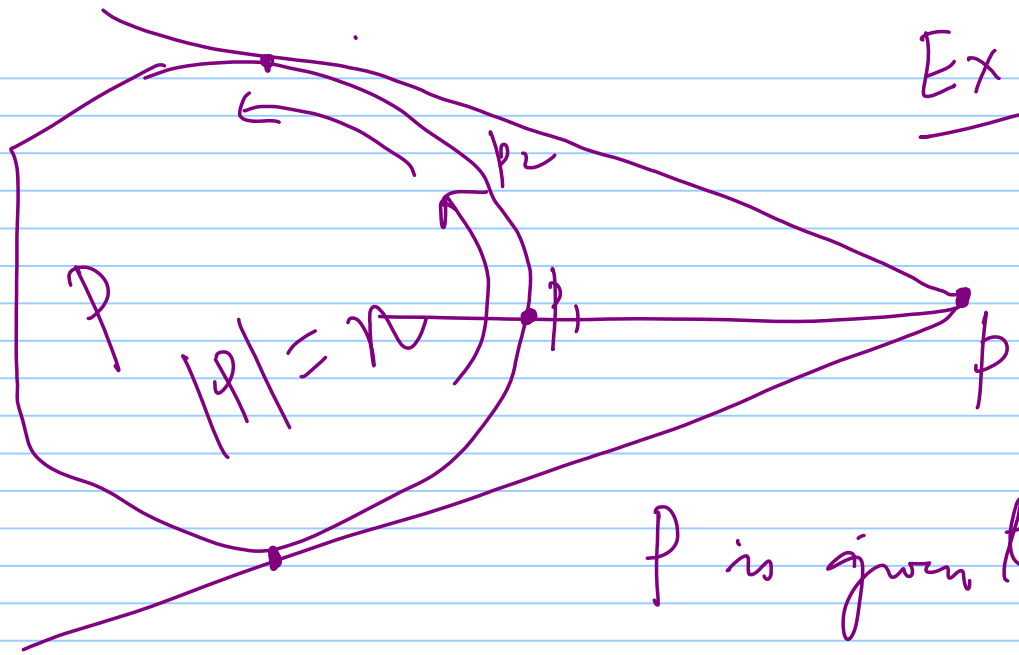
$$= \frac{r^{n+1} - 1}{r - 1}$$

$r > 1$

$$\approx O(r^n)$$



$1 + 2 + 4$



Ex: $O(\log_2 n)$

binary search

Tutorial

P is given to you in a ccw.

Input: A convex polygon P of n points and a point p.

Output: The two tangents from p to P.

Chan's algorithm:

- Point set P has n points
- Someone (an oracle) has told us the value of h.

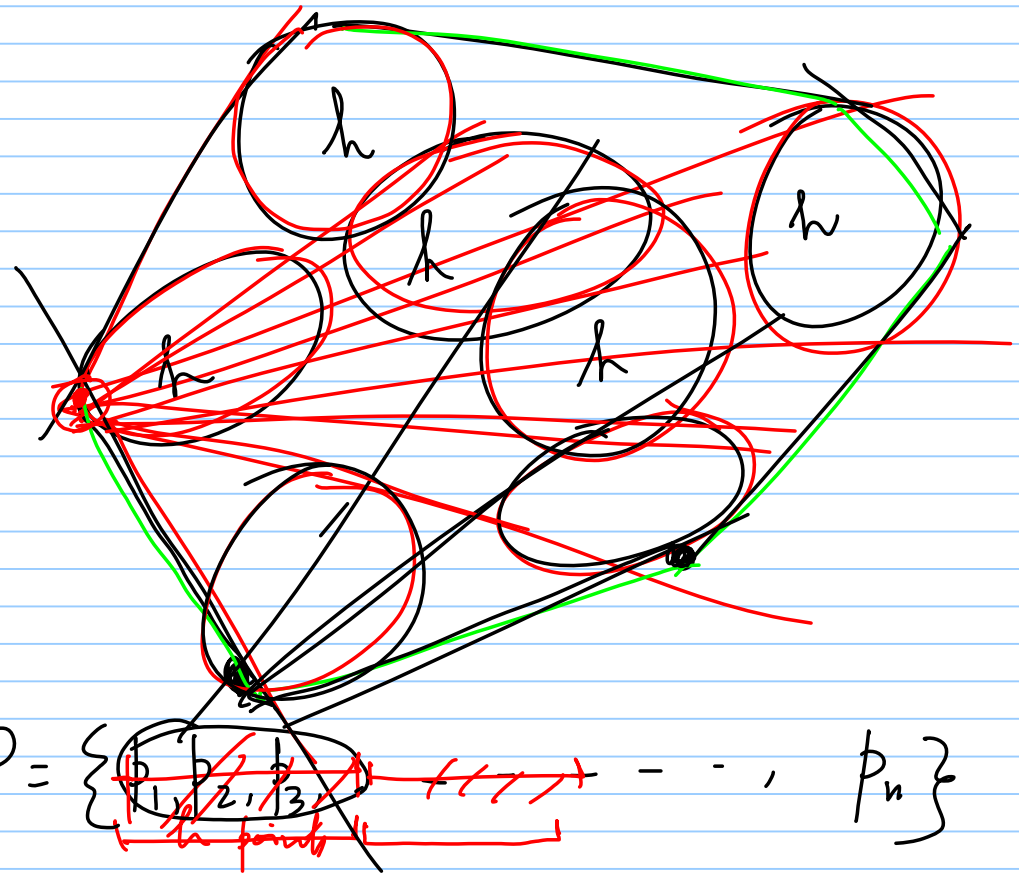
Group the points arbitrarily, so that each group has h many points.

How many groups: n/h

For each group, compute the convex hull using Graham's scan — $O(h \log h)$

So, in total, we take $O\left(\frac{n}{h} \cdot h \log h\right)$

$= O(n \log h)$



Tangent to each convex polygon — $O(\log h)$

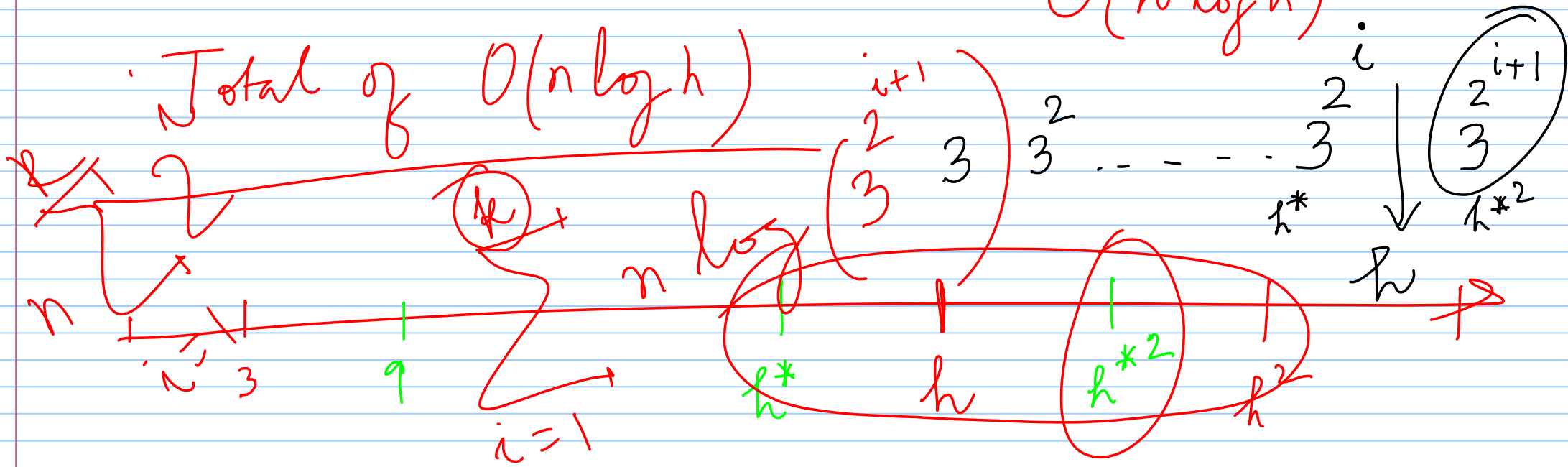
How many tangents we draw — $O(n/h)$

$$\left.\begin{matrix} O(\log h) \\ O(n/h) \end{matrix}\right\} \boxed{O\left(\frac{n}{h}\log h\right)} \times h$$

$$= O(n \log h)$$

Total of $O(n \log h)$

$$\sum_{i=1}^{h} n \log$$



$P = \{p_1, p_2, \ldots, p_n\}$

Goal: To compute the min^m of P.

Operations:

1. Comparison
2. updation.

You cannot avoid $O(n)$ comparisons.

Q: How many updates can there be?

backward analysis

$P_i = \{p_1, p_2, \ldots, p_i\}$

$\frac{1}{i} = Pr[X_i = 1]$

$$X_i = \begin{cases} 1 & \text{if there is an updation at the } i^{th} \text{ iteration} \\ 0 & \text{o/w.} \end{cases}$$

$$E[X] = E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i]$$

total # updations

$$E[X_i] = Pr[X_i = 1]$$

$$\sum_{i=1}^{n} \frac{1}{i} \approx O(\log n)$$

$$E[X] = \sum_x x\, p(x)$$

$$p(x) = Pr(X = x)$$

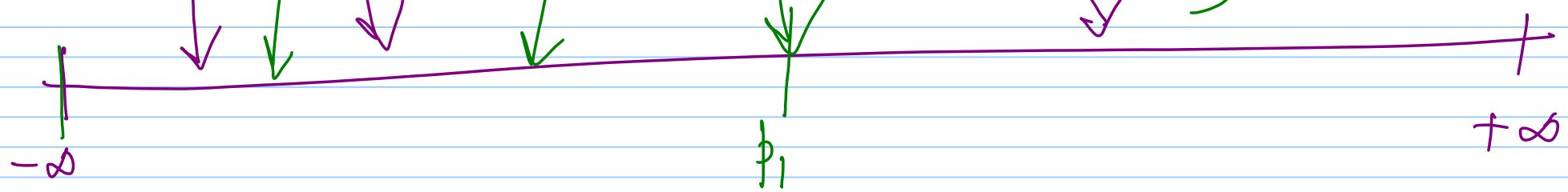
Linearity of expectations

$X_1, X_2, \ldots, X_n$

$$X = \sum_{i=1}^{n} X_i$$

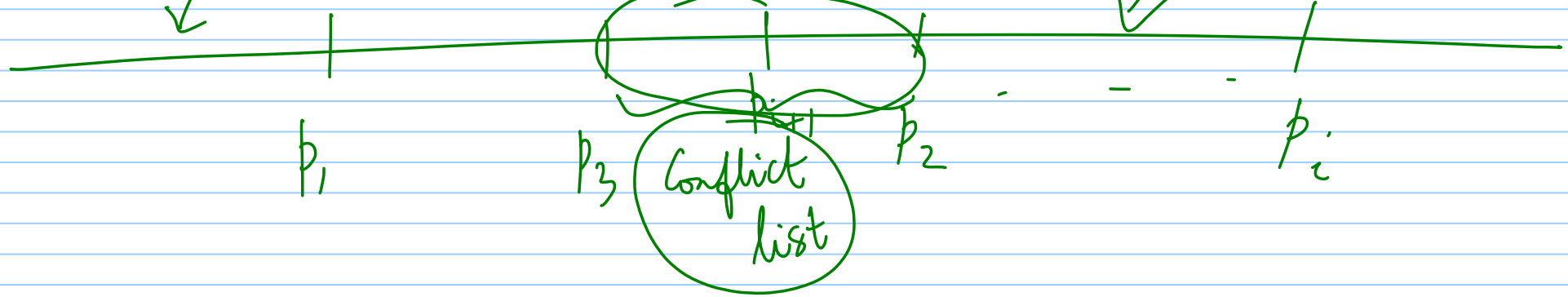
$$E[X] = E[X_1] + \cdots + E[X_n]$$

$$P = \{p_1, p_2, p_3, \ldots, p_n\}$$

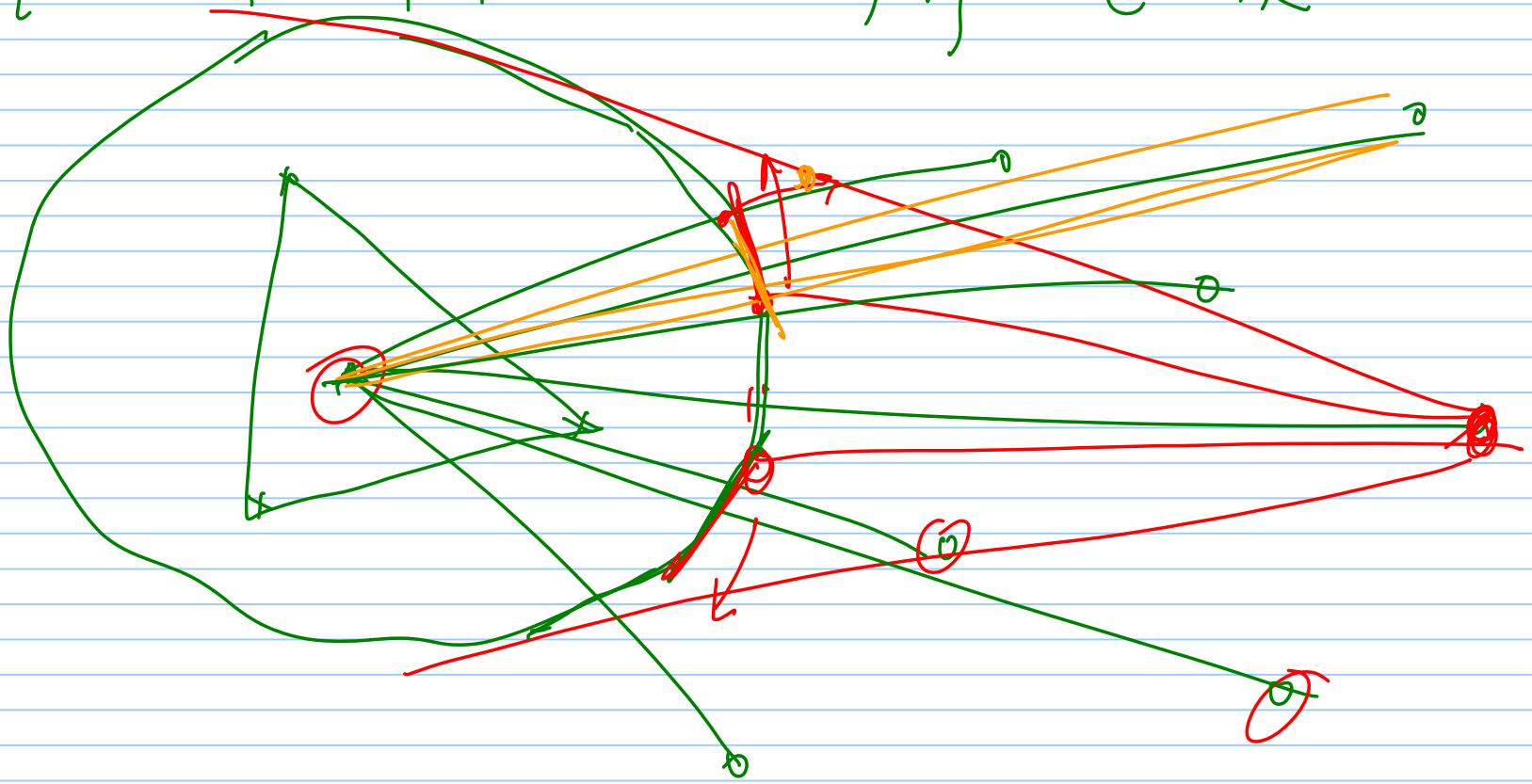


$P = \{p_1, p_2, \ldots p_i, p_{i+1}, \ldots p_n\}$

$$n \sum_i \frac{O(1)}{i} \approx O(n \log n)$$



$$P = \{p_1, p_2, \ldots, p_i, p_{i+1}, \ldots, p_n\} \in \mathbb{R}^2$$



books

- David Mount lecture notes
- Marc Overmars, Marc de Berg, Marc van Kreveld, Otfried Cheong
  - textbook for CG.
- M. I. Shamos, F. Preparata
- Jeff Erickson lecture notes.

- Ketan Mulmuley (CG through randomization)
- Motwani & Raghavan (not a full CG book)