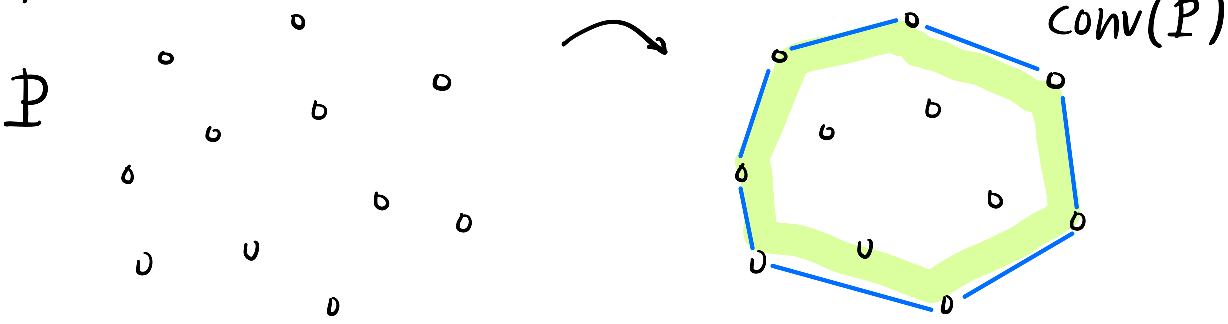
CMSC 754 - Computational Geometry
Lecture 2: Convex Hulls in the Plane

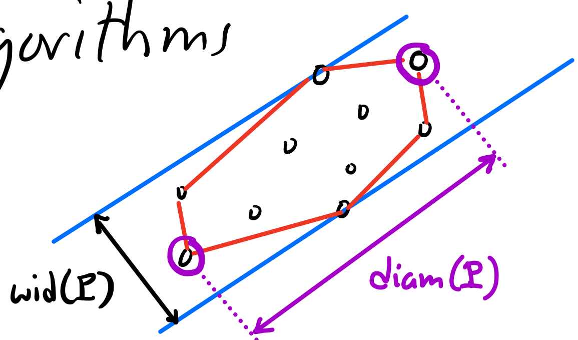
Convex Hull: (Intuitive definition)

Given a point set P in $\mathbb{R}^2$, imagine snapping a rubber band around the points



Uses:

- shape approximation (intersection test)
- first step in other algorithms
    - diameter
    - width



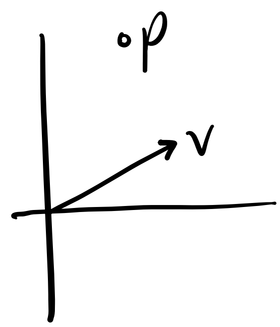
Basic Definitions:

$\mathbb{R}^d$ - Real d-dim space $p=(p_1,\dots,p_d)$ $p_i \in \mathbb{R}$

- Refer to as

points (p,q) - location

or vectors (u,v,w) - displacement



$\mathbb{R}$ - scalars $\alpha, \beta, \gamma, \dots$

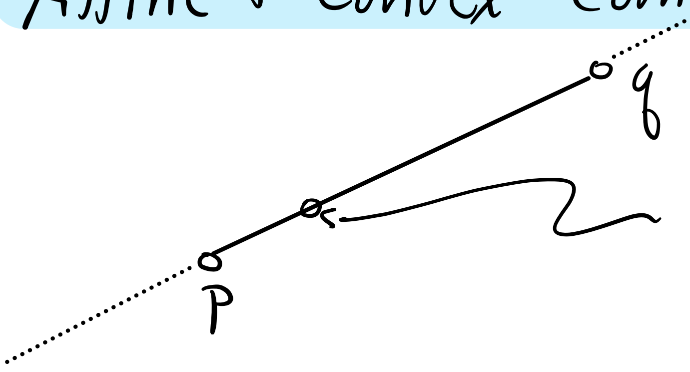
usual ops from linear algebra:

$u+v$, $u-v$ - vector addition

$\alpha \cdot u$ - scalar multiplication

$u \cdot v$ - dot product $= \sum_{i=1}^{d} u_i v_i$

Affine + Convex Combinations:



for $\alpha \in \mathbb{R}$

$(1-\alpha)p + \alpha q$

Generally given $p_1 \dots p_k$:

Affine combination: $\sum_{i=1}^{k} \alpha_i p_i$ $\sum \alpha_i = 1$

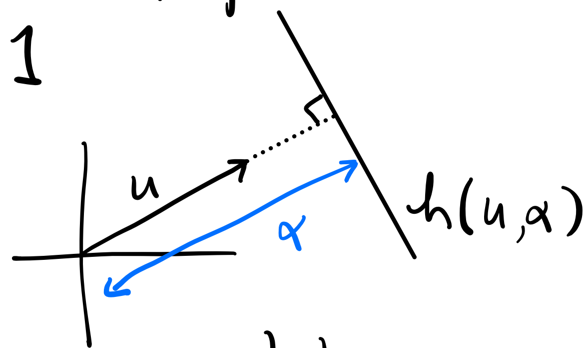
Convex combination: ... and $0 \le \alpha_i \le 1$

Lines, Hyperplanes, Halfspaces:

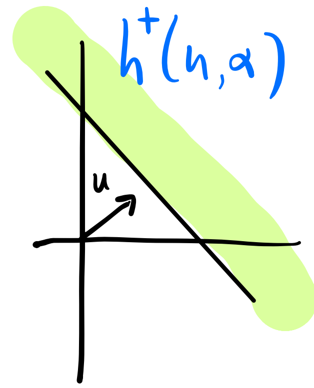
Given nonzero vector u + scalar α,

$h(u,\alpha) = \{ p \in \mathbb{R}^d \mid p \cdot u = \alpha \}$ is hyperplane

If $\|u\| = 1$



$h^+(u,\alpha) = \{ p \in \mathbb{R}^d \mid p \cdot u \geq \alpha \}$

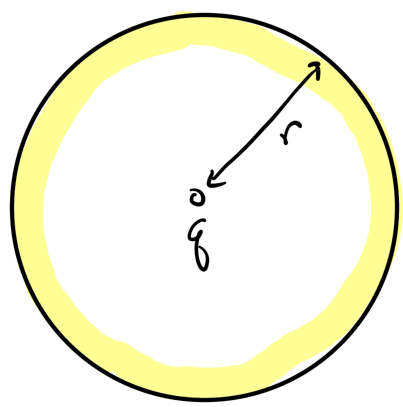


Euclidean Ball:

$\text{dist}(p,q) = \|p-q\| = \left(\sum_{i=1}^{d} (p_i - q_i)^2 \right)^{1/2}$

$B(q,r) = \{ p \in \mathbb{R}^d \mid \|p-q\| \leq r \}$

(Euclidean) ball of radius r centered at q.



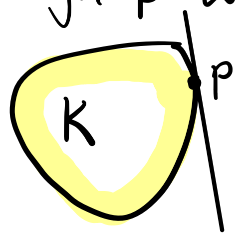
Convexity:

A set $K \subseteq \mathbb{R}^d$ is convex if $\forall p, q \in K$ the line segment $\overline{pq}$ (equiv. any conv. combination of $p + q$) lies within K

Boundary of K

Support Hyperplane:

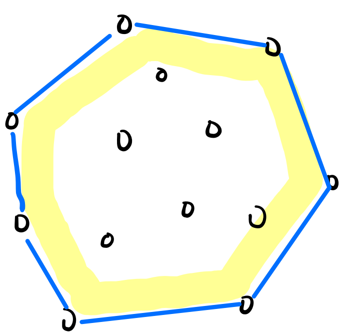
Given convex K and any point $p \in \partial K$, $\exists$ hyperplane passing through p with K lying all on one side.



Convex Hull:

Given a set P of points in $\mathbb{R}^d$, the convex hull, $\mathrm{conv}(P)$, is the smallest convex set containing P.

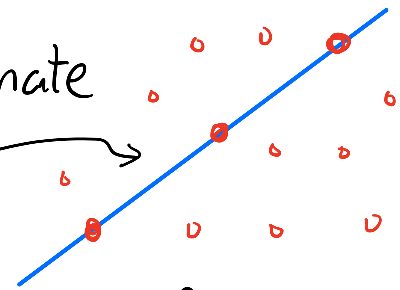
- The set of all convex combs in P
- The intersection of all halfspaces containing P



General Position:

Geometric algorithms are complicated by rare (?) degenerate cases:

- points having same coordinate
- ≥ 3 collinear points
- ≥ 4 cocircular points



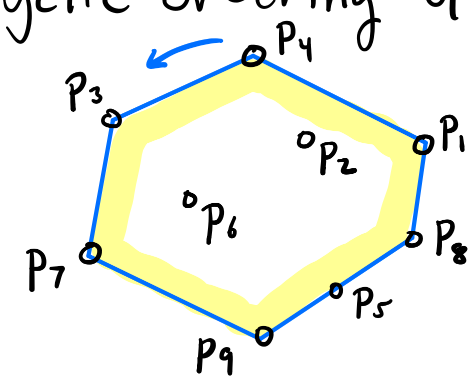
To simplify algorithm presentation we often assume these do not arise in the input.

Called **general-position assumption**

(Planar) Convex Hull Problem:

Given a set of n pts $P = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^2$ ($p_i = (x_i, y_i)$) compute $\mathrm{conv}(P)$.

Output: Cyclic ordering of vertices on the hull



possible output: (indices)

$\langle 4, 3, 7, 9, 8, 1 \rangle$

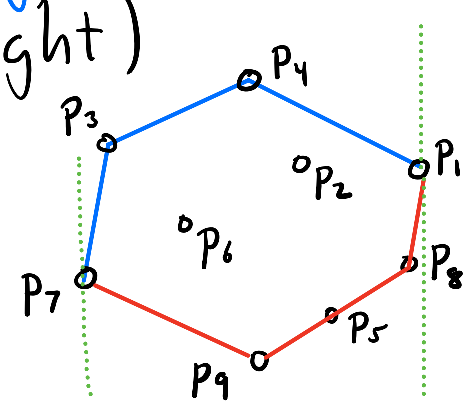
Note: p_5 not output

(Can assume this away by "general position")

Alternative output: (left to right)

Upper-hull + **Lower-hull**

$\langle 7, 3, 4, 1 \rangle + \langle 7, 9, 8, 1 \rangle$

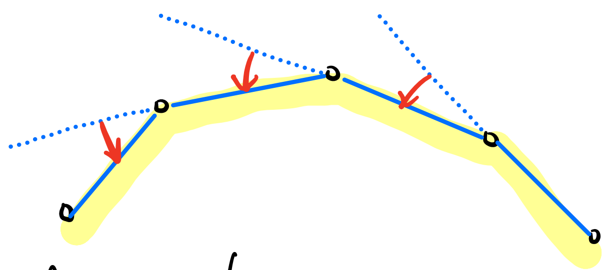


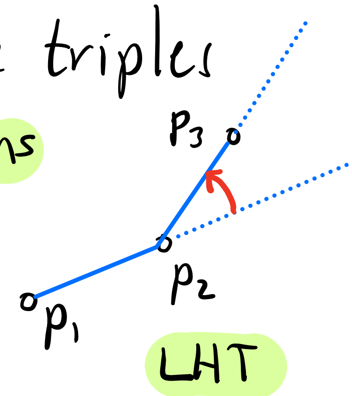
Graham's Scan: $O(n \log n)$ solution

- Compute upper + lower hulls separately
- Upper-hull:
  - Sort pts by x-coords
  - Add each to upper hull
  - Remove pts no longer on hull ← How?
- Lower-hull: (symmetrical)

Observations:

- The rightmost pt always on hull
- Reading right to left, consecutive triples on the hull form left-hand turns

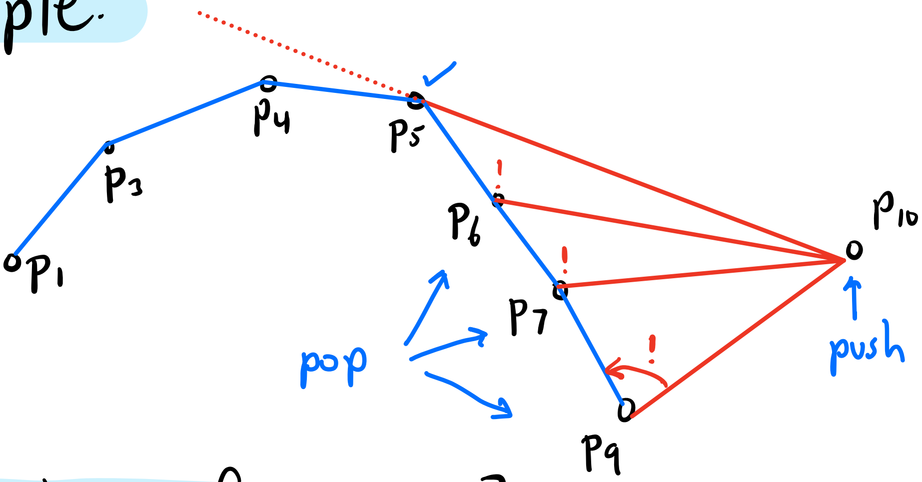


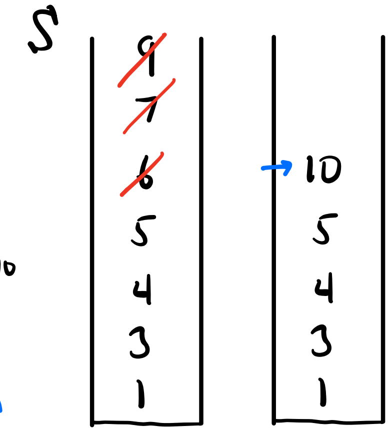


Incremental Approach:

- Store vertices (indices) of upper hull on stack
- For each new point p_i (left to right)
  - While $\langle p_i, S[top], S[top-1] \rangle$ do not form LHT — pop S
  - Push p_i

Example:





How to test for LHT?

Orientation test

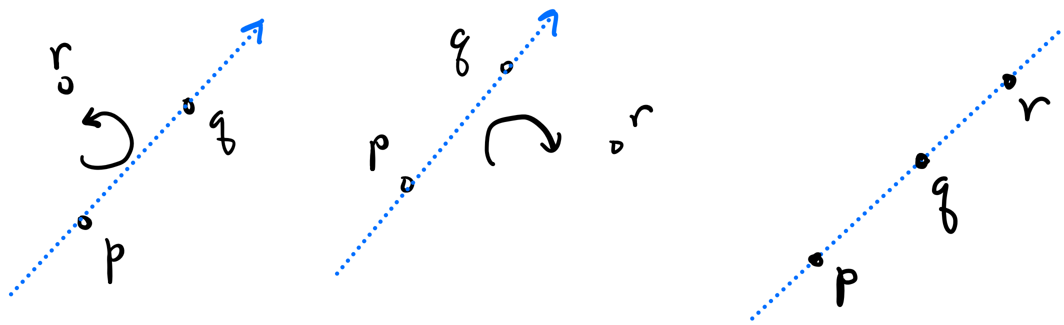
Given a sequence $\langle p, q, r \rangle$ of 3 pts in $\mathbb{R}^2$

$$\text{orient}(p,q,r) = \text{sign}\left(\det\begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}\right)$$

is: $+1$ if they are oriented CCW (LHT)

-1 " " " " CW (RHT)

0 if they are collinear (or duplicates)



+1   -1   0

Graham's Scan: (Upper Hull only)

- Sort pts by increasing x-coords $\langle p_1 \ldots p_n \rangle$
- Push p_1 + p_2 onto S
- for $i \leftarrow 3$ to n
  - while $\big(|S| \geq 2$ and $\text{orient}(p_i, S[t], S[t-1]) \leq 0\big)$ pop S

    t = "top"
  - push p_i

Correctness:

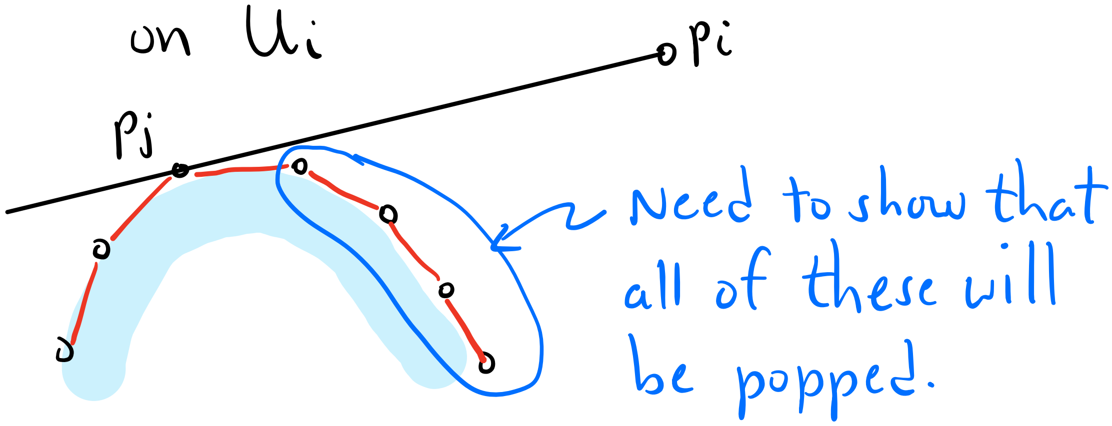
Lemma: After processing p_i, S contains upper hull of $\langle p \ldots p_i \rangle$

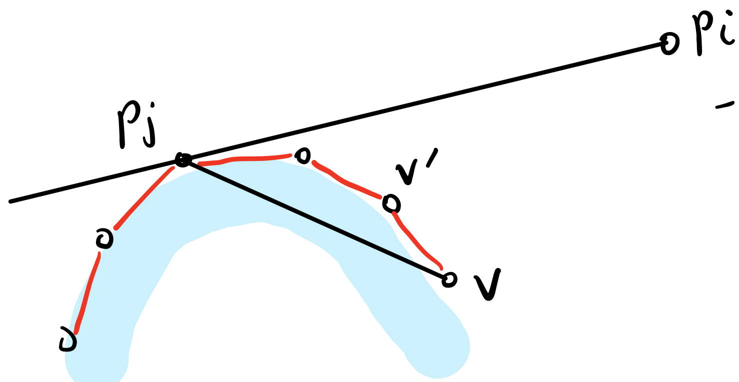
Proof: By induction on i.

Basis: $i = 1, 2$ – Trivial

Step: For $i \geq 3$, let U_i denote the vertices on upper hull up to p_i.

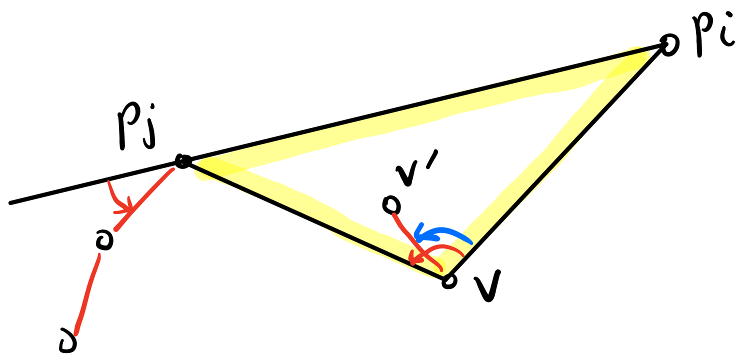
- By induction U_{i-1} is correct up to p_{i-1}
- We'll show its correct after adding p_i
- Let p_j denote vertex before p_i on U_i





- Let v be current vertex + let v' be predecessor

- By convexity:
  - all pts lie below $\overline{p_i p_j}$
  - all pts after p_j lie above $\overline{p_j v}$



$\Rightarrow$ v' lies in $\Delta p_j v p_i$

$\Rightarrow$ $\angle p_i v v' \leq \angle p_i v p_j \leq 2\pi$

$\Rightarrow$ $\text{orient}(p_i, v, v') \leq 0$

$\Rightarrow$ v is popped off stack

On arriving at p_j, orientation flips so popping stops at p_j + finally p_i pushed $\square$

Running time:

- $O(n \log n)$ to sort
- for $3 \le i \le n$, let d_i = num. of pops when inserting p_i
- Time for scan is ~

$$\sum_{i=3}^{n} (\underset{\substack{\uparrow \\ \text{for pops}}}{d_i} + \underset{\substack{\nwarrow \\ \text{for push of } p_i}}{1}) \le n + \sum_{i=3}^{n} d_i$$

- Note that $\sum d_i \le n$ → Why?

- Total time: $O(n \log n + 2n) = O(n \log n)$

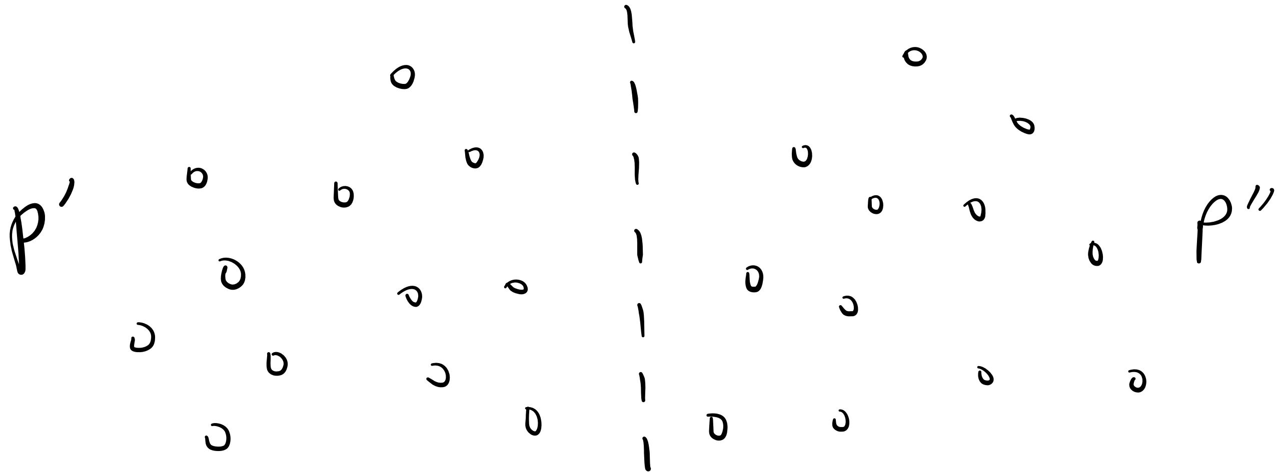
Divide + Conquer Algorithm:

Given point set $\underline{P}$:

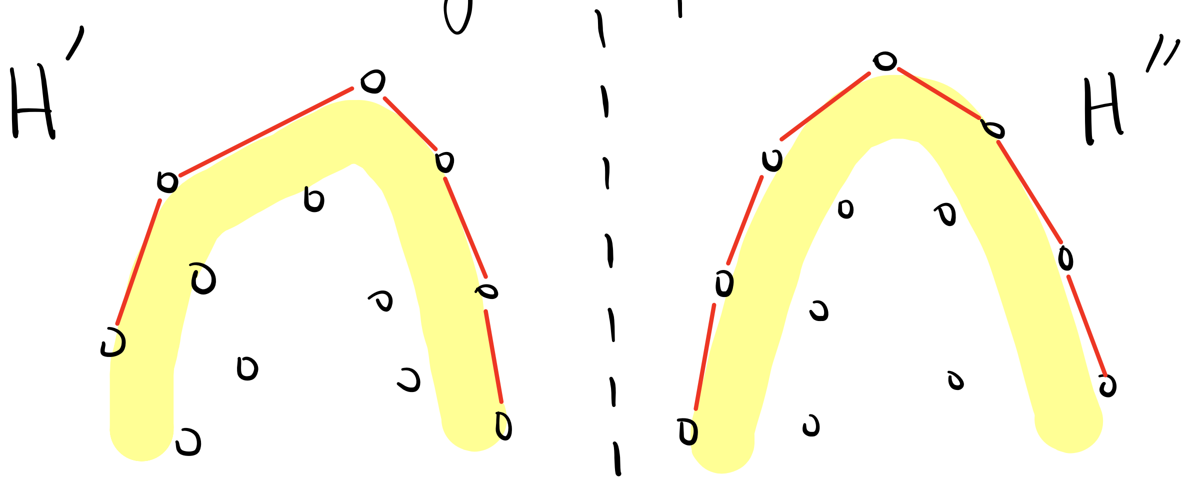
if $|\underline{P}| \le 3$ then compute hull by brute force ($O(1)$)

else

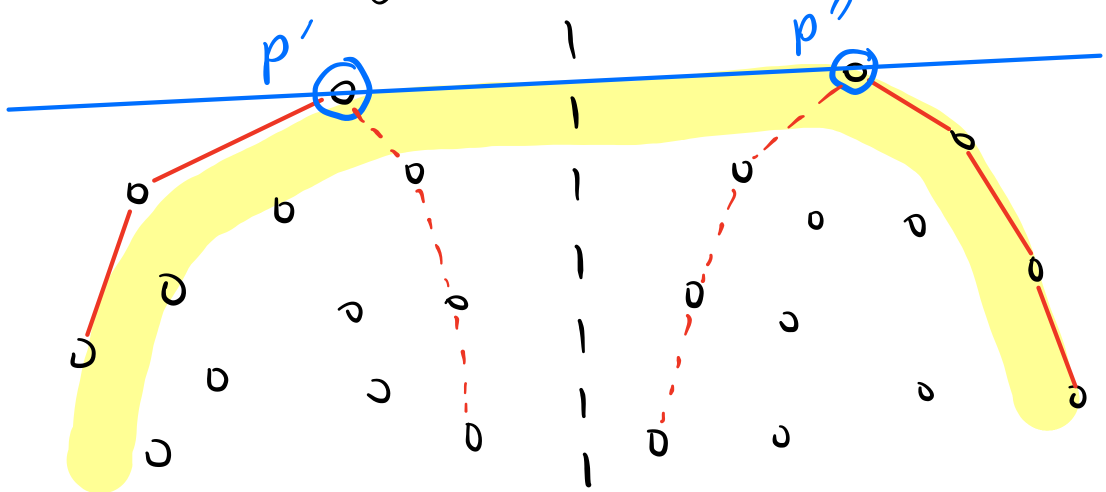
- Partition $\underline{P}$ by vertical line into $\underline{P}', P''$ of sizes ~ $n/2$



- Recursively compute hull of each



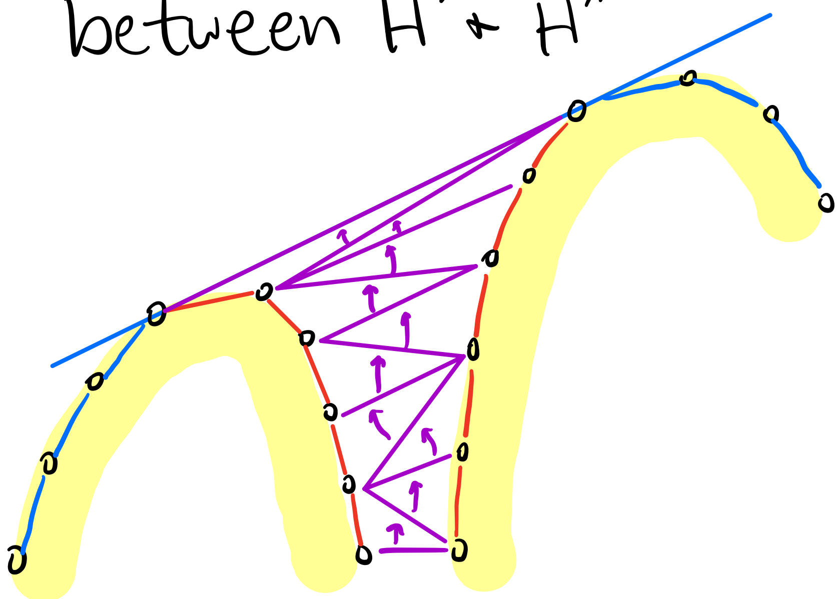
- Compute pts $p' \in H'$ + $p'' \in H''$ defining upper tangent



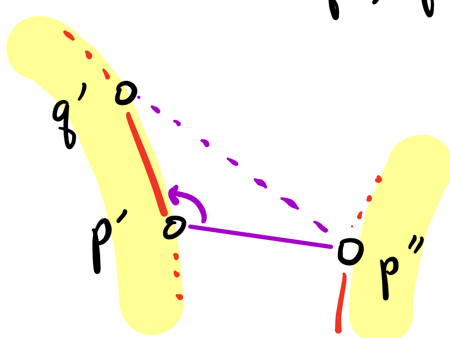
- Merge partial hulls together

How to compute upper tangent?

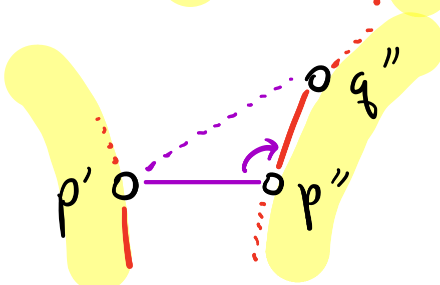
- Start with a chord joining closest points (w.r.t. x) of H' + H''
- "**Walk**" this chord up the ladder between H' + H''



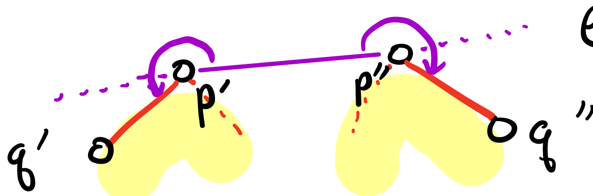
- **How?** Let p', p'' be current vertices
  Let q', q'' be vertices above



if $\text{orient}(p'', p', q') \le 0$
$p' \leftarrow q'$



if $\text{orient}(p', p'', q'') \ge 0$
$p'' \leftarrow q''$



else <u>done</u>!

Correctness? (Exercise)

Running time?

- $O(n)$ time to find upper tangent by walking
  - Each step takes $O(1)$ time
  - Eliminates one vertex
- Gives recurrence:

$$T(n) = \underbrace{2T(n/2)}_{\text{Recursively compute two hulls, each from } n/2 \text{ pts}} + \underbrace{n}_{\text{Split, tangent, merge}}$$

- Same as Mergesort. By Master Theorem

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

Jarvis March: An $O(nh)$ algorithm

Idea: Compute any one vertex of hull $\rightarrow v_1$

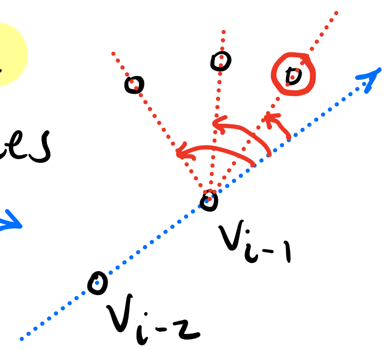
for $i = 2, 3, \ldots$

  compute next vertex v_i on hull

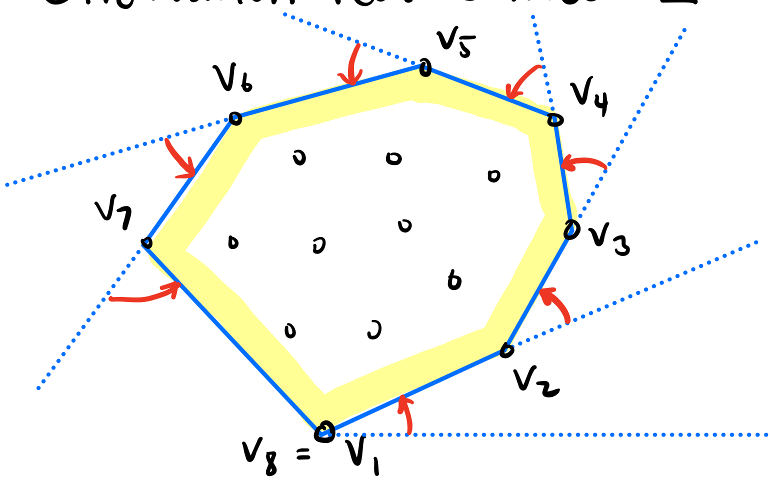
  if ($v_i == v_1$) return $\langle v_1, \ldots, v_{i-1} \rangle$

v_1? Point of P with min y-coordinate

next vertex? The point of P that minimizes turn angle w.r.t. prior two vertices



[This doesn't require trig. Orientation test suffices]



Correctness: Easy

Running time: Compute v_1 - $O(n)$

Compute v_i - $O(n)$ ← Repeat h times

Total: $O((h+1)n) = O(h \cdot n)$