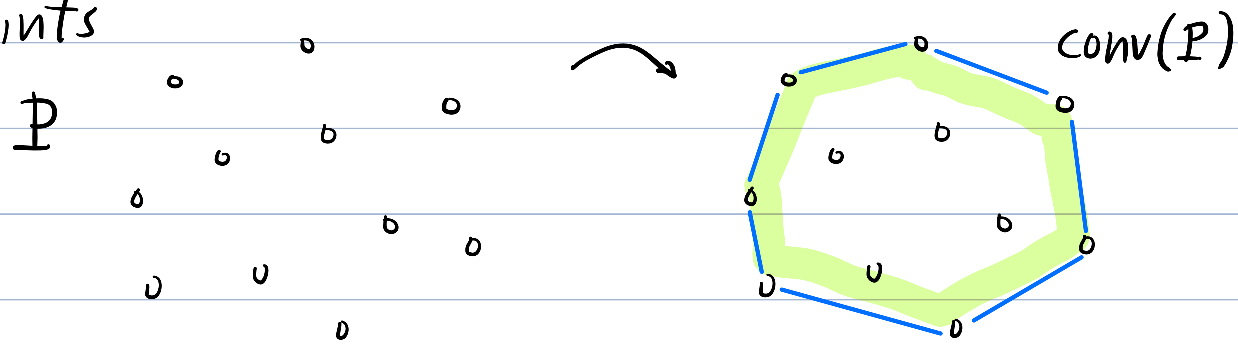
CMSC 754 - Computational Geometry
Lecture 2: Convex Hulls in the Plane

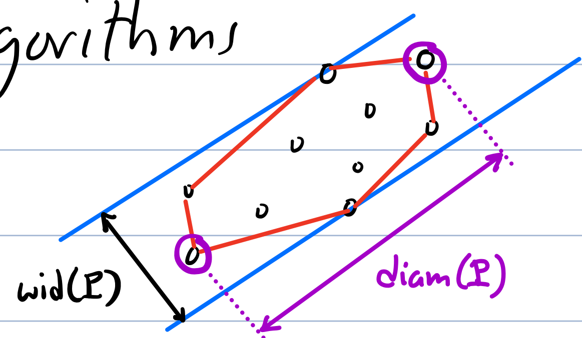
Convex Hull: (Intuitive definition)

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



Geometric Basics:

Affine Geometry: The geometry of "flats"

Objects:

- **Scalars** (Real numbers) $3, -17, \pi, \sqrt{2}$
- **Points** (Position) $p \cdot \quad \cdot q$
- **(Free) Vectors** (Direction + magnitude)
  ↳ not anchored to origin ↗ ↘

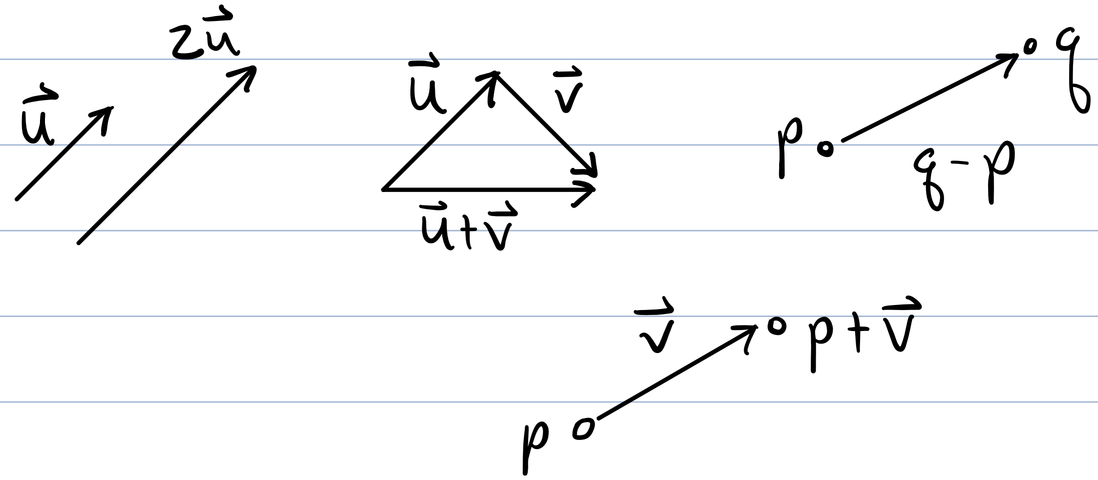
Operations:

$S \cdot V \to V$ - scalar-vector mult

$V + V \to V$ - vector addition (+ subtract)

$P - P \to V$ - point-point subtract

$P + V \to P$ - point-vector addition



Note: $P + P$ not allowed

Affine Combination:

Given points p, q and scalars α, β where $\alpha + \beta = 1$, define:

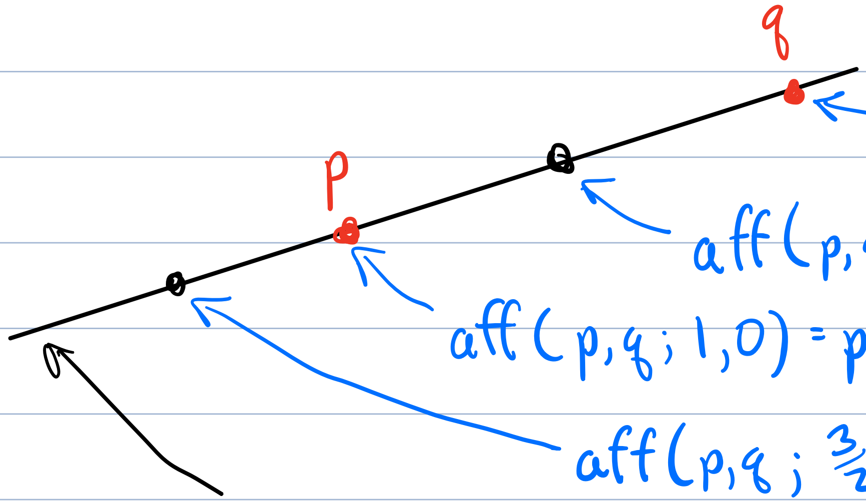
$$\mathrm{aff}(p, q; \alpha, \beta) = p + \beta(q - p)$$

Equivalent to: $p + \beta q - \beta p$

$$= (1-\beta)p + \beta q$$

$$= \alpha p + \beta q$$

Weighted sum of $p + q$



aff(p,q; 0,1) = q

aff(p,q; ½, ½) = midpoint

aff(p,q; 1,0) = p

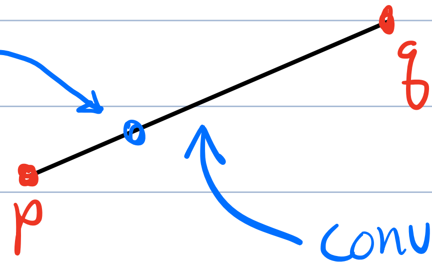
aff(p,q; 3/2, -1/2)

line $\overleftrightarrow{pq}$ is affine closure of p+q

Convex Combination:

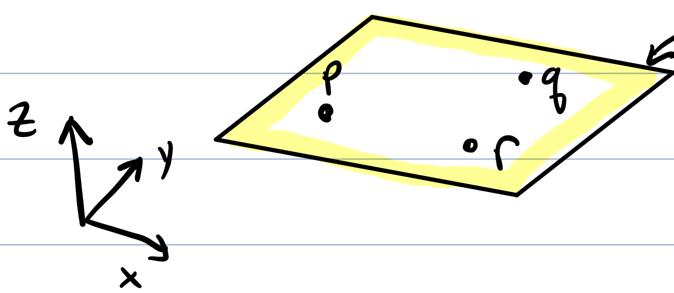
Special case when $0 \le \alpha, \beta \le 1$

$\frac{2}{3}p + \frac{1}{3}q$

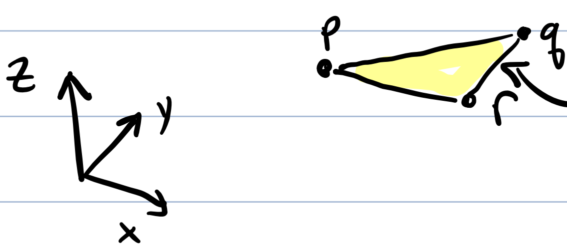


convex combinations generate all points on segment $\overline{pq}$

These generalize to multiple points:



affine closure of p,q,r in $\mathbb{R}^3$ is the plane through these pts.



convex closure is Δpqr

But... No way to define angles or lengths!

Euclidean Geometry: Affine + Inner product

Inner product - Linear operator that maps 2 vectors to a scalar

$$\langle \circ, \circ \rangle : V \times V \rightarrow S$$

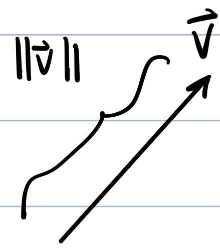
Dot product (most common inner product)

$$\vec{u} = (u_1, \dots, u_d) \quad \vec{v} = (v_1, \dots, v_d)$$

$$\vec{u} \cdot \vec{v} = u_1 v_1 + \dots + u_d v_d = \sum_{i=1}^{d} u_i v_i$$

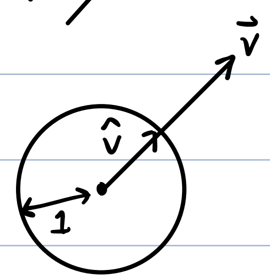
Inner product allows us to define:

Vector length: $\|\vec{v}\| = \sqrt{\vec{v} \cdot \vec{v}}$



Normalization (to unit length):

$$\hat{v} = \vec{v} / \|\vec{v}\|$$

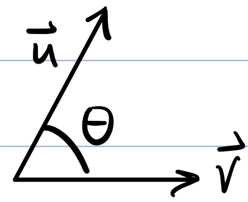


Distance between points:

$$\mathrm{dist}(p,q) = \|q - p\|$$

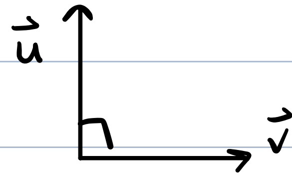
Angles between vectors:

$$\mathrm{ang}(\vec{u}, \vec{v}) = \cos^{-1}(\hat{u}, \hat{v})$$



Perpendicularity:

$$\vec{u} \perp \vec{v} \Leftrightarrow \vec{u} \cdot \vec{v} = 0$$

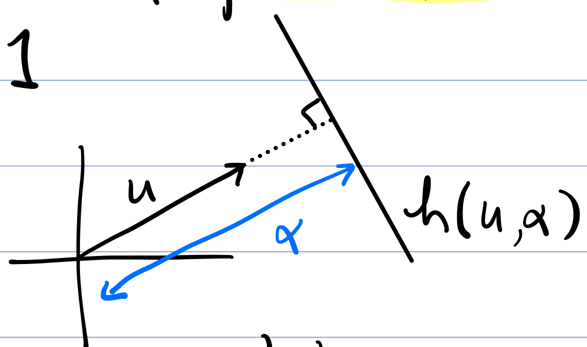


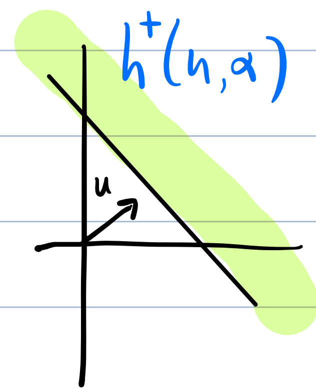
Lines, Hyperplanes, Halfspaces:

Given nonzero vector u + scalar α,

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

If $\|u\| = 1$

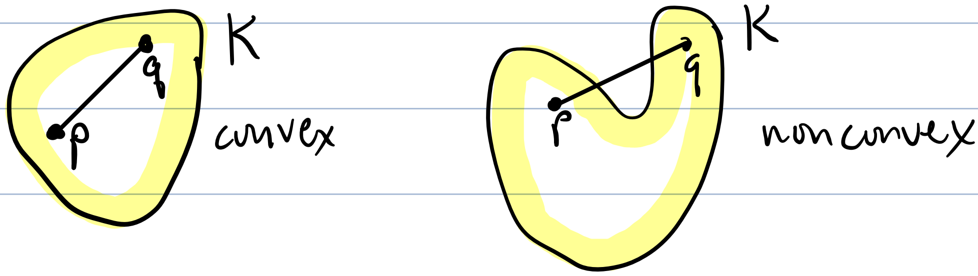




$$h^+(u,\alpha) = \{ p \in \mathbb{R}^d \mid p \cdot u \geq \alpha \}$$

Convexity:

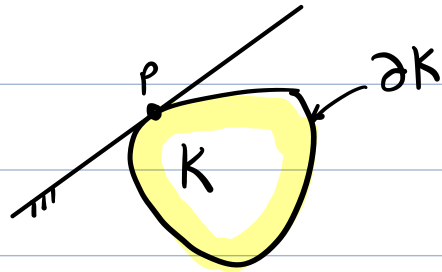
A set $K \subseteq \mathbb{R}^d$ is convex if $\forall p, q \in K$ the line segment $\overline{pq}$ (equiv. any conv. combination of p + q) lies within K



Support Hyperplane:

Boundary of K

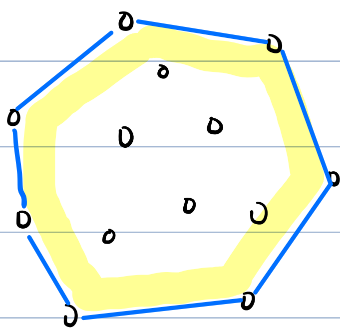
Given convex K and any point $p \in \partial K$,
$\exists$ hyperplane passing through p with K
lying all on one side.

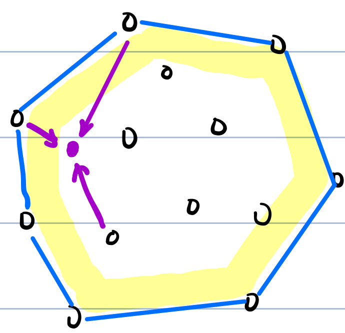


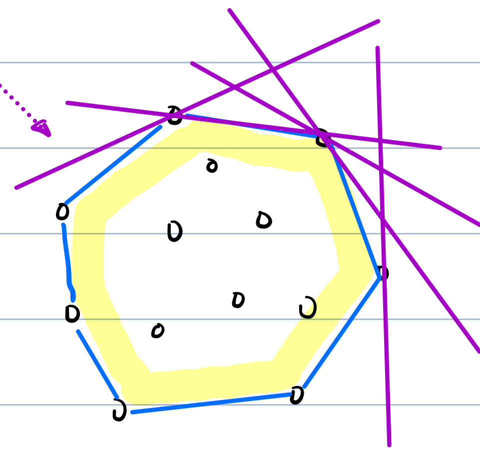
Convex Hull:

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

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



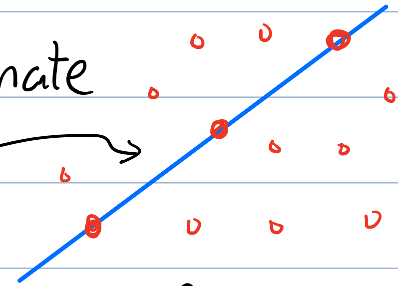




General Position:

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

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



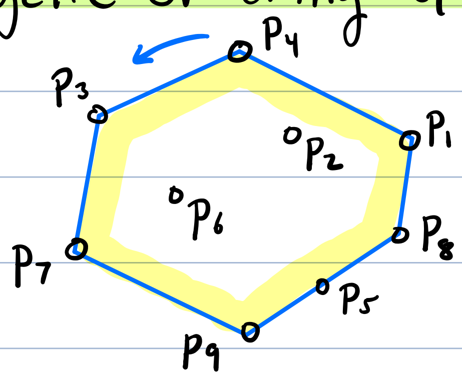
To simplify algorithm presentation we often assume these do not arise in the input.

Called general-position assumption

(Planar) Convex Hull Problem:

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

Output: Cyclic ordering of vertices on the hull



possible output: (indices)

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

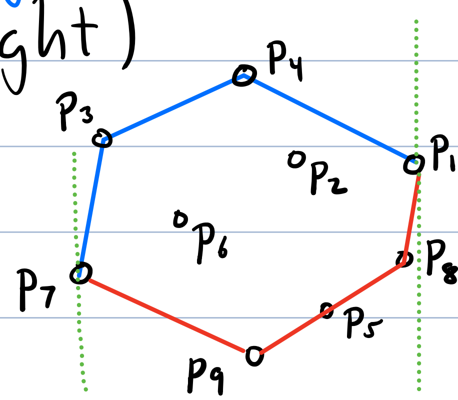
Note: p_5 not output

(Can assume this away by "general position")

Alternative output: (left to right)

Upper-hull + Lower-hull

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

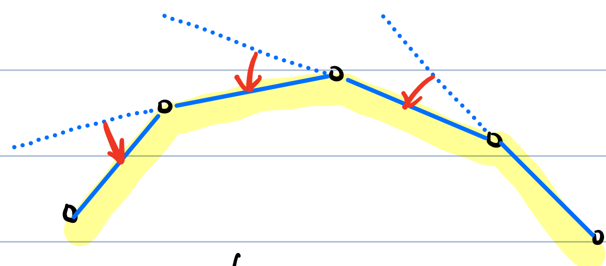


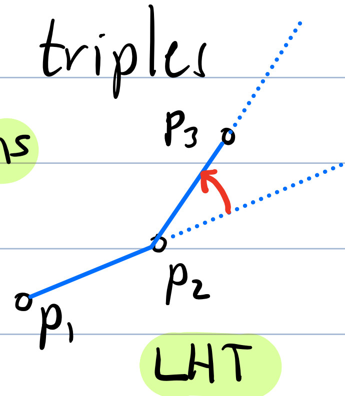
Graham's Scan: $O(n \log n)$ solution

- 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

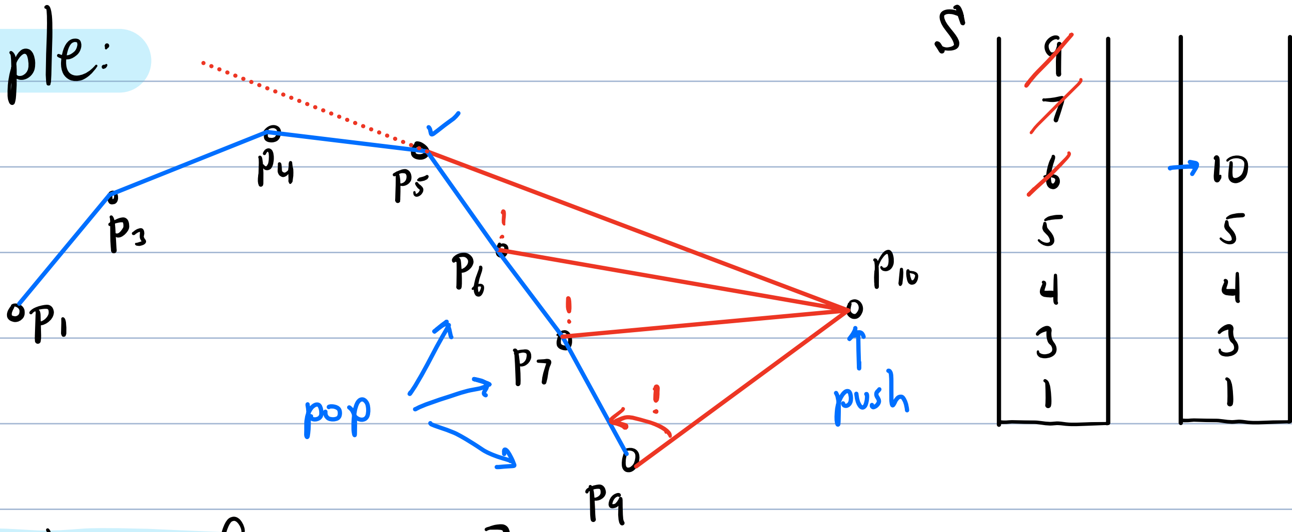




Incremental Approach:

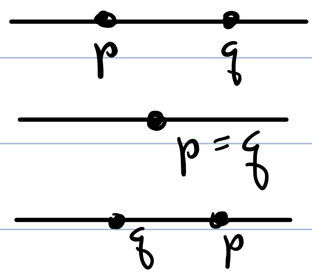
- Store vertices (indices) of upper hull on stack
- For each new point p_i (left to right)
  - While $\langle p_i, S[top], S[top-1] \rangle$ do not form LHT — pop S
  - Push p_i

Example:



How to test for LHT?

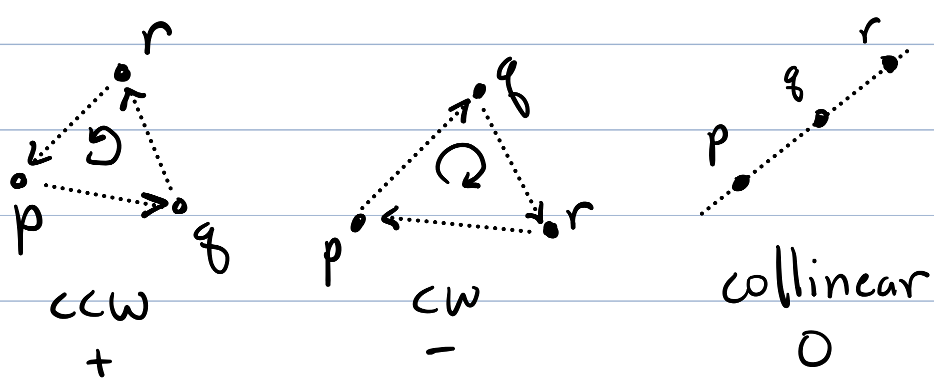
Orientation: In $\mathbb{R}^1$, for any p, q we have either:



$p < q$	$q - p > 0$	$+$
$p = q$	$q - p = 0$	0
$p > q$	$q - p < 0$	$-$

How to define "order" in $\mathbb{R}^2$??

Ans: order = orientation is a property of a seq. of 3 points



How to compute?

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

Note: Order of points is critical!

Orientation can generally be defined on any sequence of $d+1$ points in $\mathbb{R}^d$

Graham's Scan: (Upper Hull only)

- Sort pts by increasing x-coords $\langle p_1 \dots p_n \rangle$
- 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 :

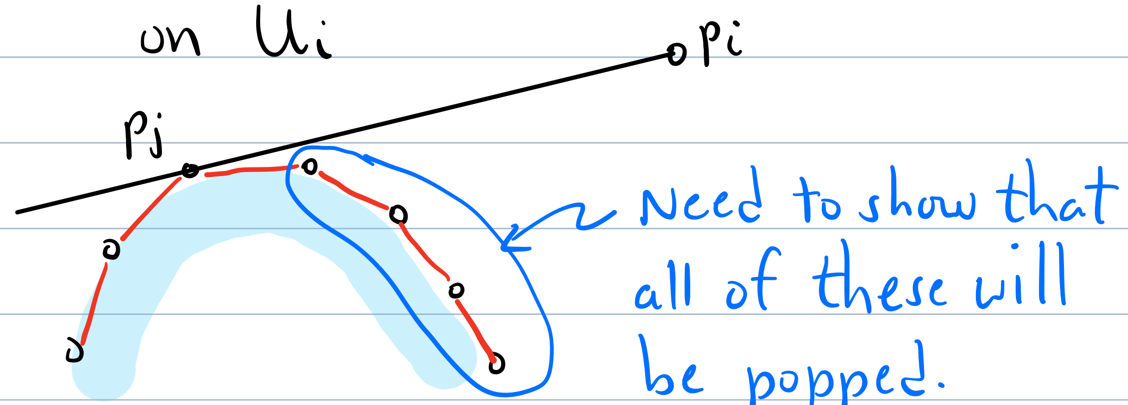
Lemma: After processing p_i, S contains upper hull of $\langle p \dots p_i \rangle$

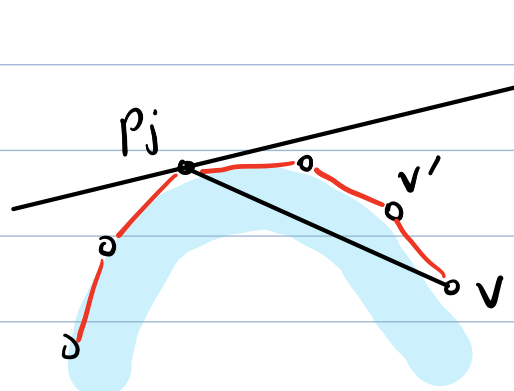
Proof: By induction on i.

Basis: $i = 1, 2$ – Trivial

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

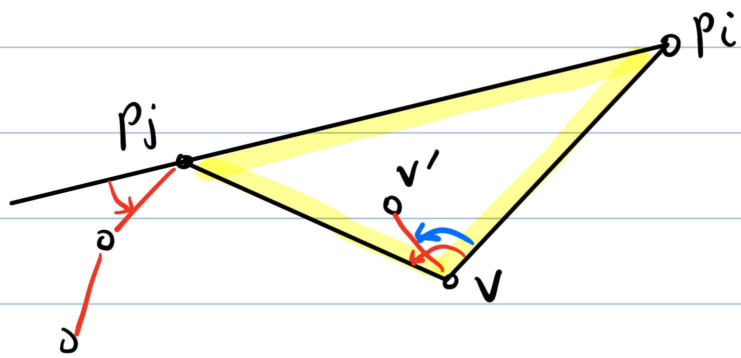
- By induction U_{i-1} is correct up to p_{i-1}
- We'll show its correct after adding p_i
- Let p_j denote vertex before p_i on U_i





- 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 $\Delta p_j v p_i$

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

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

$\Rightarrow$ 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 ~

$$\sum_{i=3}^{n} (\underbrace{d_i}_{\text{for pops}} + \underbrace{1}_{\text{for push of } p_i}) \le n + \sum_{i=3}^{n} d_i$$

- Note that $\sum d_i \le n$ → Why?

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

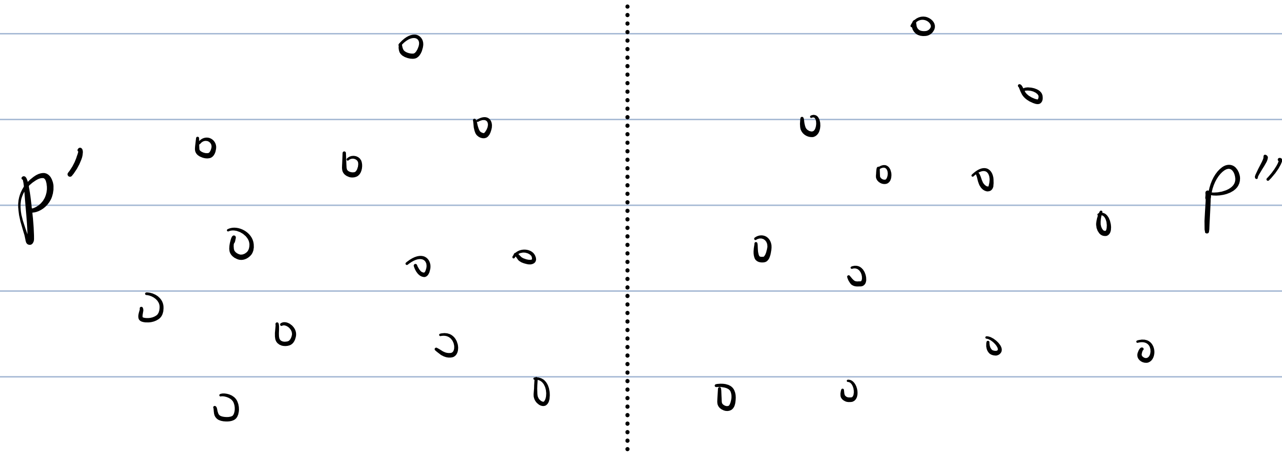
Divide + Conquer Algorithm:

Given point set $\underline{P}$:

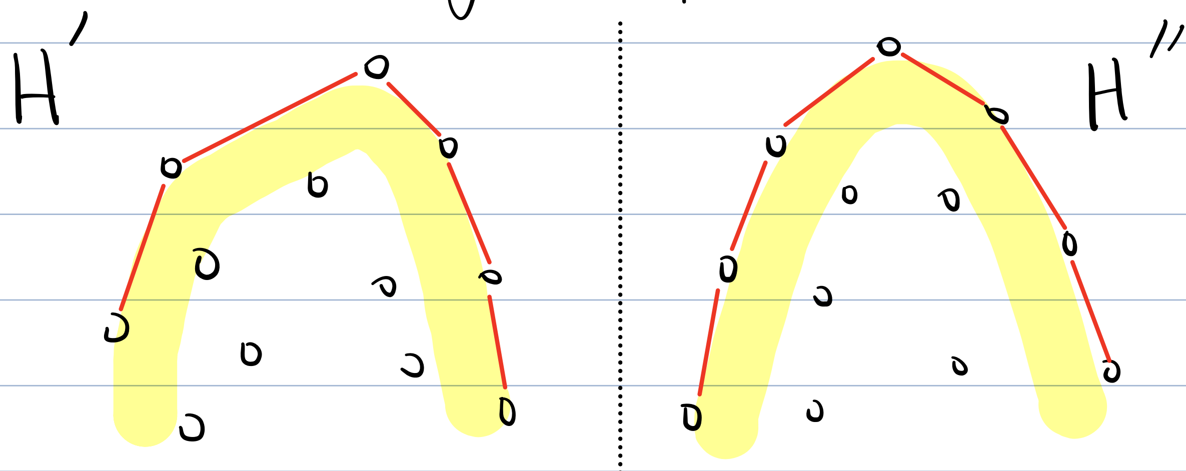
if $|P| \le 3$ then compute hull by brute force ($O(1)$)

else

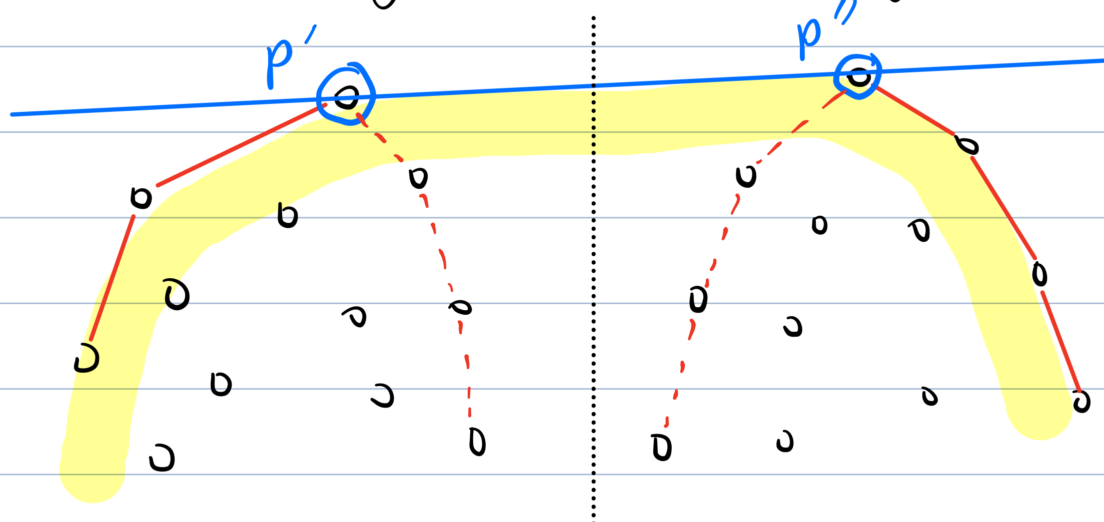
- Partition $\underline{P}$ by vertical line into $\overline{P}', P''$ of sizes $\sim n/2$



- Recursively compute hull of each



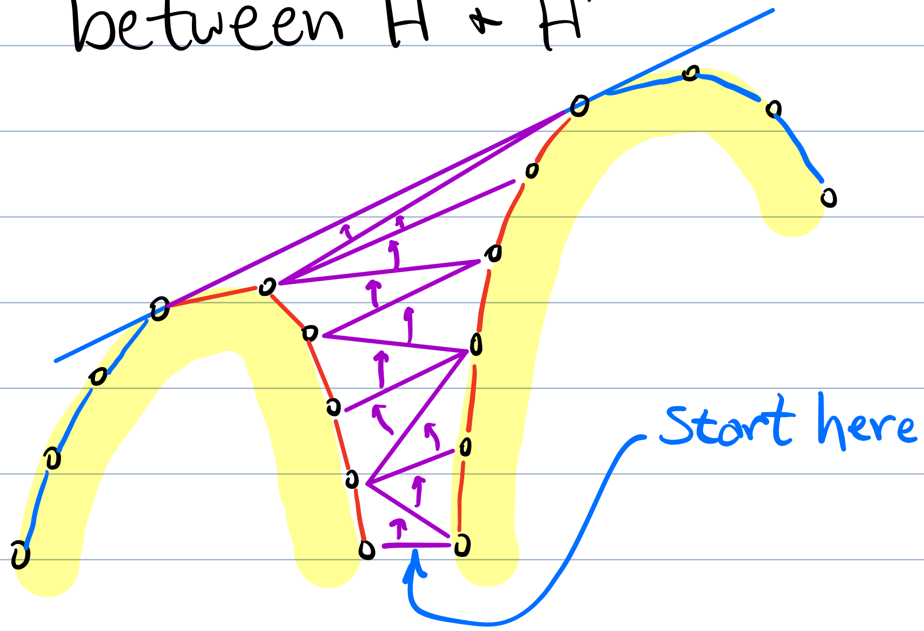
- Compute pts $p' \in H'$ + $p'' \in H''$ defining upper tangent



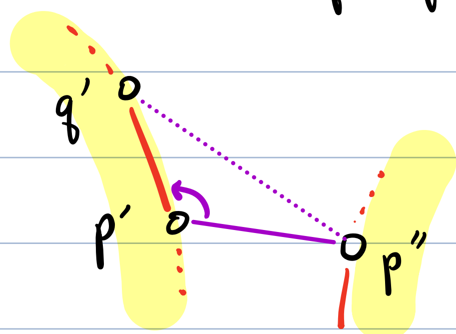
- Merge partial hulls together

How to compute upper tangent?

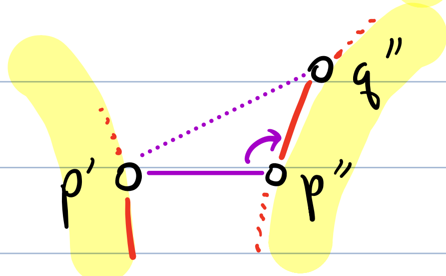
- Start with a chord joining closest points (w.r.t. x) of H' + H''
- "Walk" this chord up the ladder between H' + H''



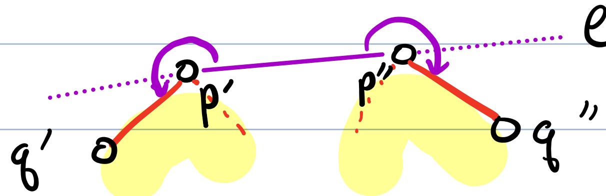
- How? Let p', p'' be current vertices
  Let q', q'' be vertices above



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



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



else done!

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 $\to v_1$

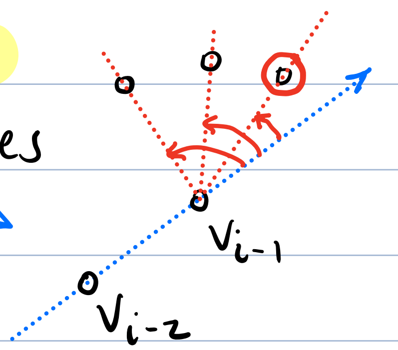
for $i = 2, 3, \ldots$

compute next vertex v_i on hull

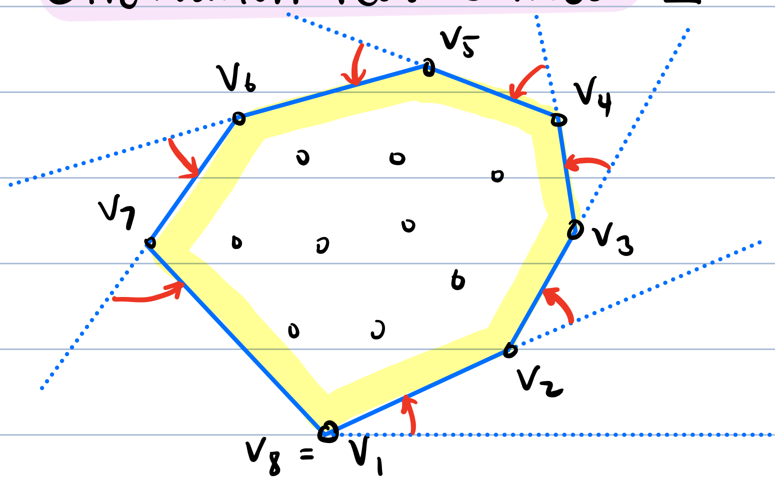
if $(v_i == v_1)$ return $\langle v_1, \ldots, v_{i-1} \rangle$

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] ← (Exercise)



Correctness: Easy

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

Compute v_i - $O(n)$ ← Repeat h times

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

Summary:

- Affine geometry + Orientations
- Convex hulls in $\mathbb{R}^2$
  - Graham Scan
  - Divide + Conquer
  - Jarvis March – $O(h \cdot n)$

$\left.\begin{matrix}\text{Graham Scan}\\ \text{Divide + Conquer}\end{matrix}\right\} O(n \log n)$

h = size of hull