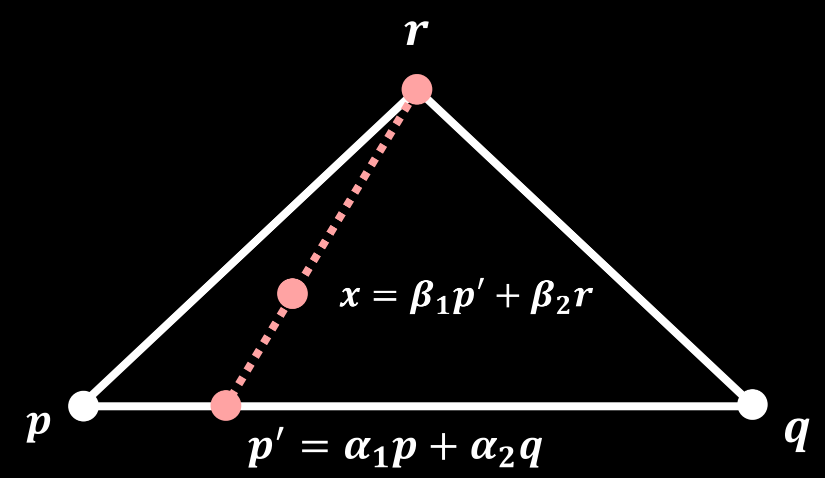
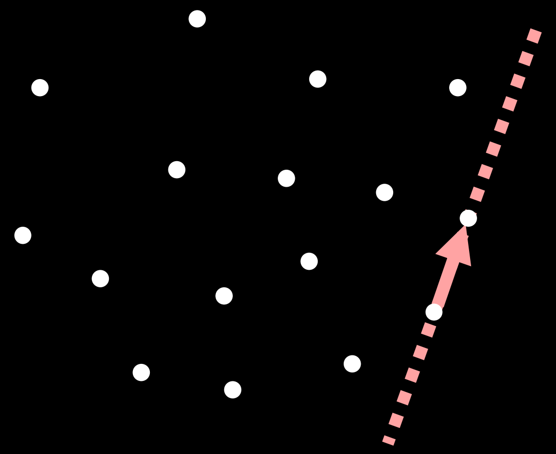
Lecture 21: Computational Geometry

Fundamental tools and the convex hull



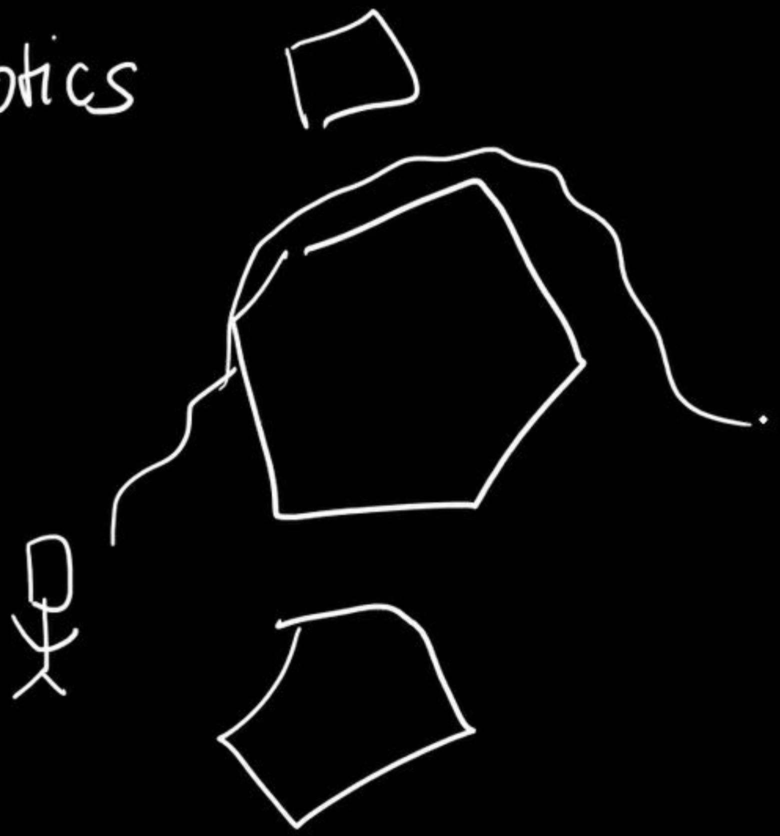


Goals for today

- Explore some fundamental tools for **computational geometry**
- Understand important tools/ideas such as:
  - **Dot and cross products**
  - The **line-side test**
  - **Convex combinations**
- Define and solve the **convex hull** problem

Why geometry?

Robotics



Representation and Model

How might we represent some of the following ideas?

Real number	Floating-point
Point	Pair of real number
Line	Equation, two points
Line Segment	Two points
Triangle	Three points

Concerns? Rounding errors. Ignore! Real arithmetic

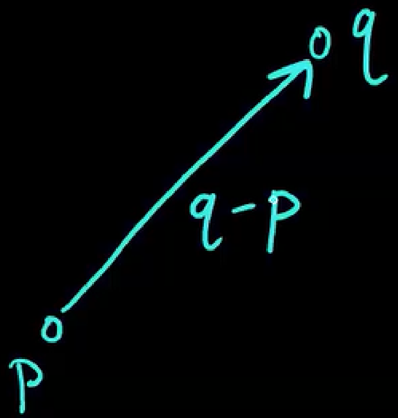
Fundamental Objects & Operations

Representation (Point): A pair of real numbers (x, y)

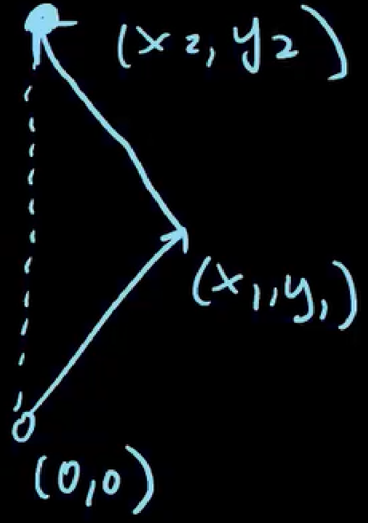
Representation (Vector): A pair of real numbers

We will use these interchangeably

Operation (Addition/subtraction):



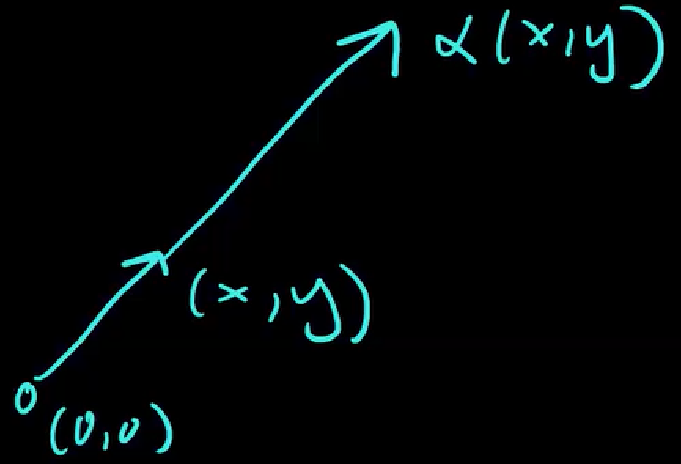
$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$$



Fundamental Operations (continued)

Operation (Scalar multiplication):

$$\alpha(x, y) = (\alpha x, \alpha y)$$



Operation (Length/magnitude):

$$\|(x, y)\| = \sqrt{x^2 + y^2}$$

App: Distance $\quad d(p, q) = \|q - p\|$

Fundamental Operations (continued)

Operation (The dot product):

$$(x_1, y_1) \cdot (x_2, y_2) = x_1 x_2 + y_1 y_2$$

Applications (Angles)

Given u, v

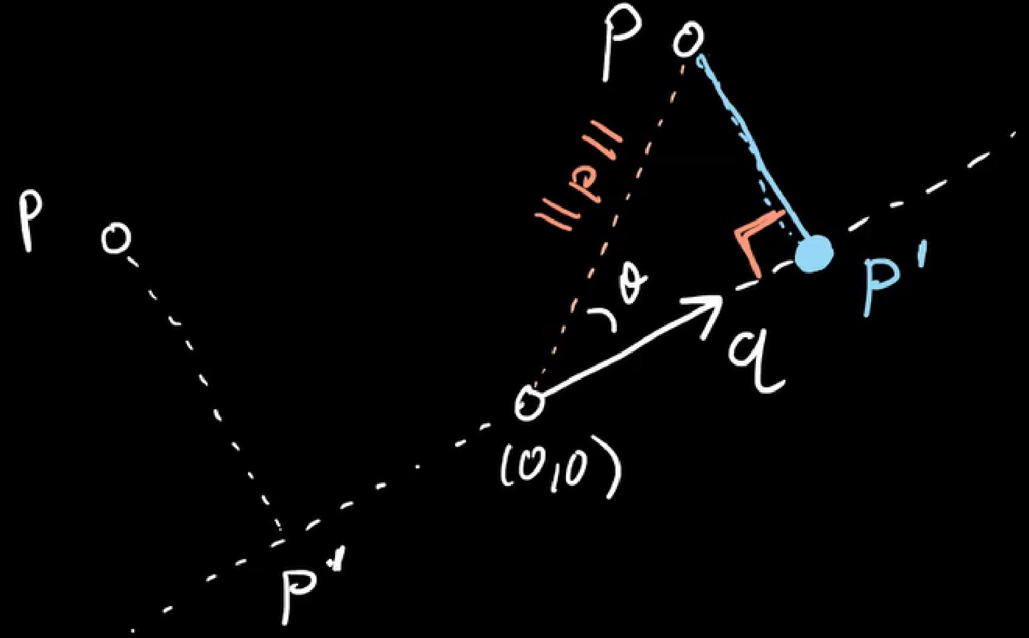
$$u \cdot v = \|u\| \|v\| \cos(\theta)$$

Application of the dot product

Application (Projection): Given a **point $\boldsymbol{p}$** and a **line $\boldsymbol{L}$** that goes through the origin in the direction of q (a unit vector), find the point p' on L that is closest to p

$$p \cdot q = \|p\| \, \|q\| \cos(\theta)$$
$$= \|p\| \cos(\theta)$$

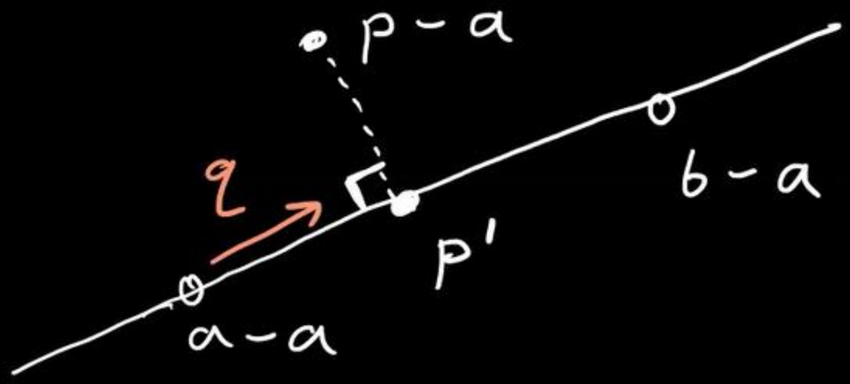
$$p' = \underbrace{(p \cdot q)}_{\text{scalar}} \, \underbrace{q}_{\text{vector}}$$



Application of the dot product

Application (Projection, but more general): Now suppose L might not go through the origin, but is defined by two points a, b on the line

$$q = \frac{b-a}{\|b-a\|} \quad (\text{unit vector})$$

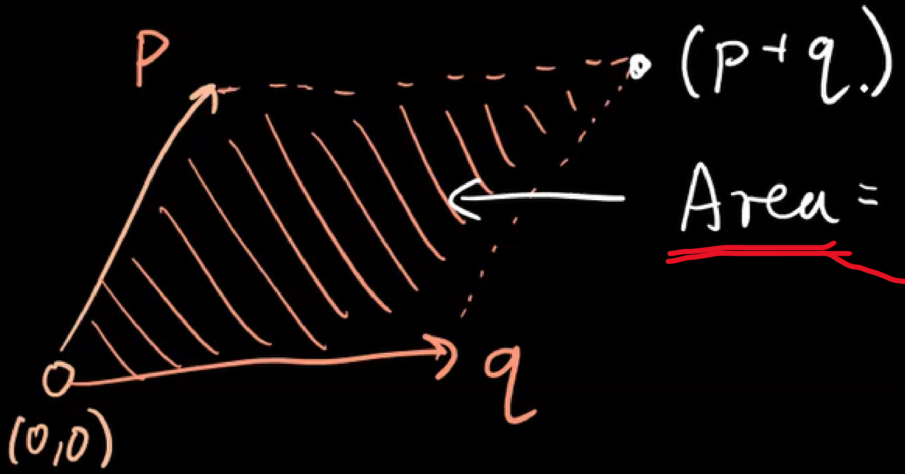


$$p' = ((p-a)\cdot q)\, q + a$$

Fundamental Operations (continued)

Operation (The cross product):

$$(x_1, y_1) \times (x_2, y_2) = \det\left(\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}\right) = x_1 y_2 - x_2 y_1$$



Area = $p \times q$

"Signed area"

negative area

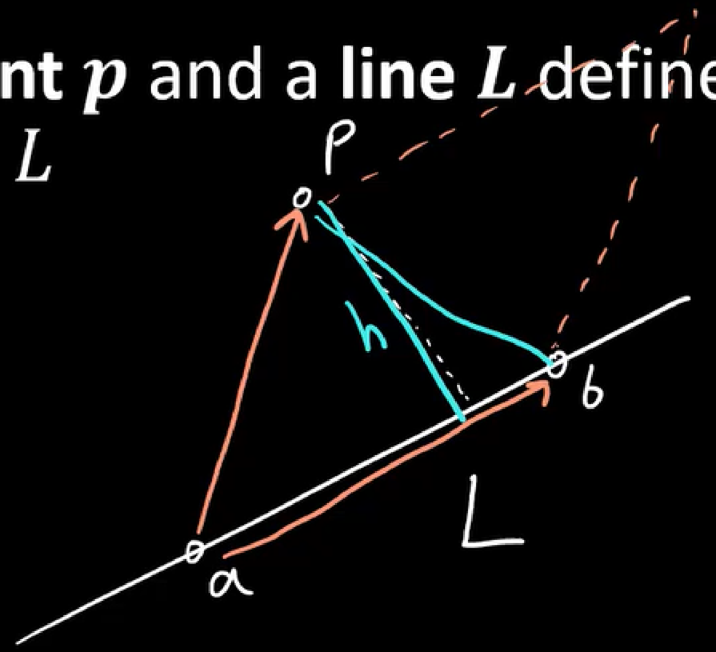
positive if q is left of p

negative otherwise.

Application of the cross product

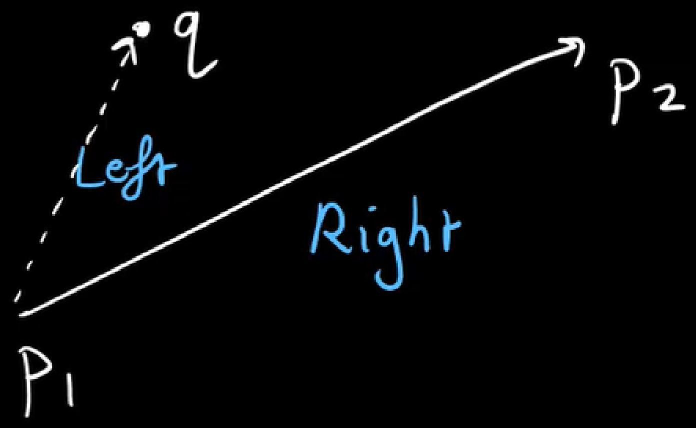
Application (Projection distance): Given a **point $\boldsymbol{p}$** and a **line $\boldsymbol{L}$** defined by two points a, b, find the distance from p to L

$$\left| \frac{(p-a) \times (b-a)}{\|b-a\|} \right|$$



Line-side test (Important!)

Operation (Line-side test): Given points p_1, p_2, q, we want to know whether q is on the LEFT or RIGHT of the line from p_1 to p_2



$$V_1 = p_2 - p_1$$

$$V_2 = q - p_1$$

$V_1 \times V_2 > 0$ LEFT

$V_1 \times V_2 < 0$ RIGHT

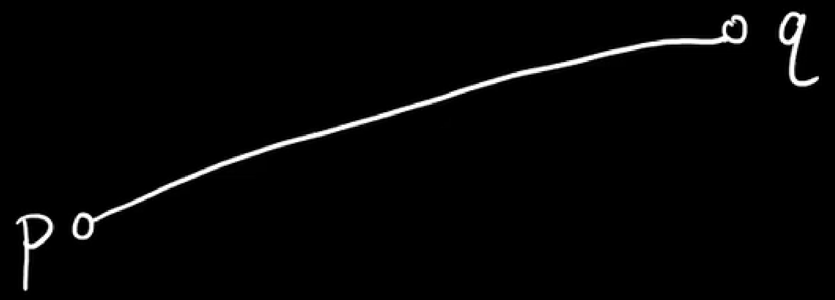
$V_1 \times V_2 = 0$ ON

Convex Combinations

Definition (Convex combination): A ***convex combination*** of the points $p_1, p_2, \dots, p_k$ is a point

$$p' = \sum_{i=1}^{k} \alpha_i \, p_i$$

such that $\sum \alpha_i = 1$ and $\alpha_i \geq 0$ for all i

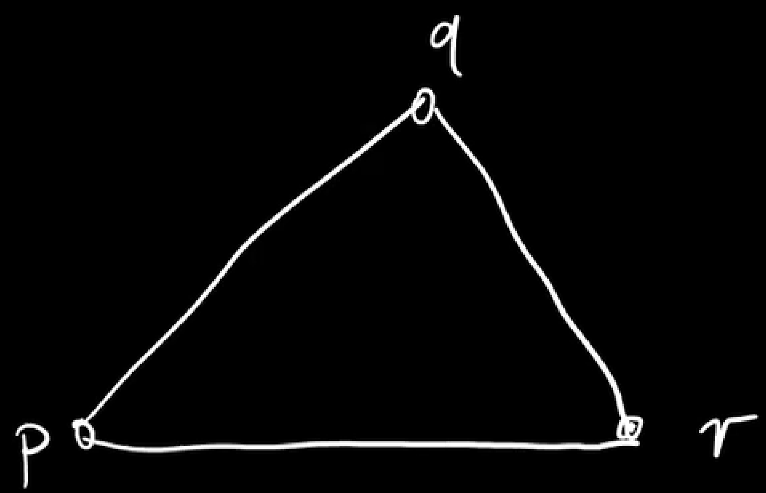


$\alpha_1 p + \alpha_2 q = p'$

Convex Combinations (continued)

Claim (Convex combination of three points): Given three points p, q, r, convex combinations of them fill the triangle with vertices p, q, r

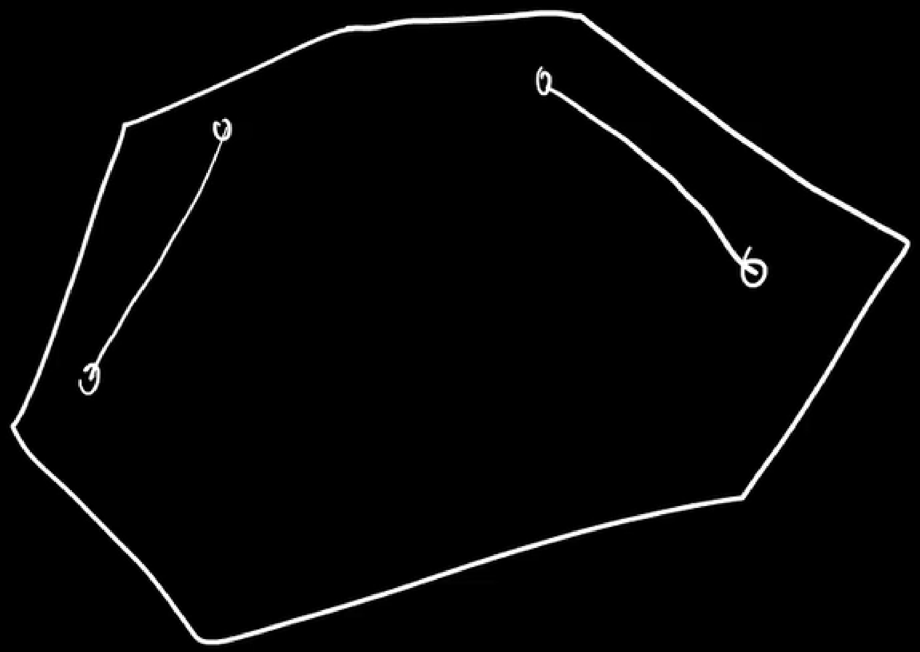
Proof as an exercise for you. See solution in the notes.

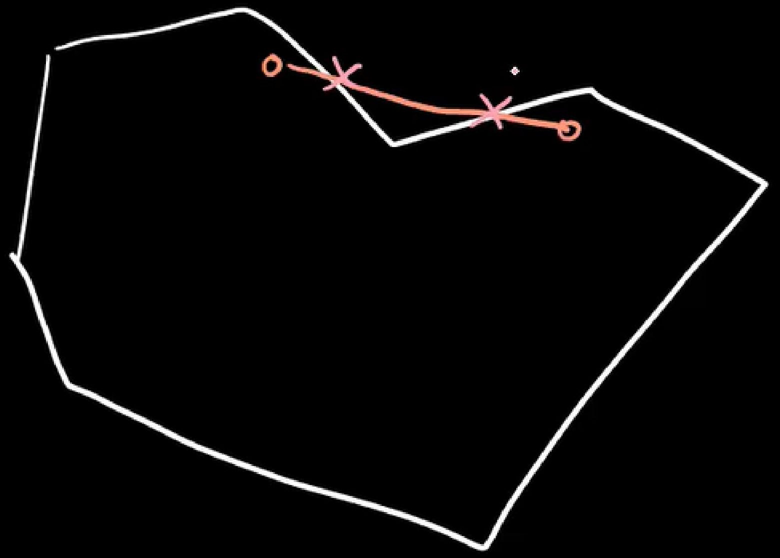


The Convex Hull

Convexity recap

Definition (Convex set): A set is convex if for any points p, q, any convex combination of p, q is also in the set

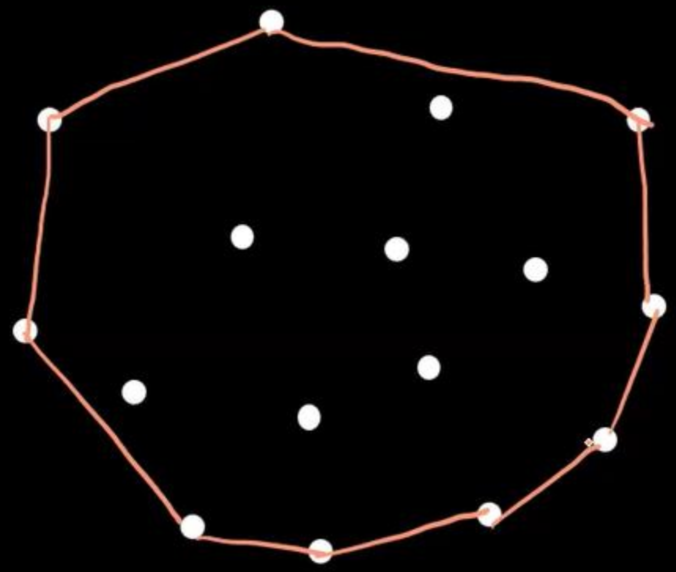




The Convex Hull

Definition (Convex hull): Given a set of points $p_1, \ldots, p_n$, the ***convex hull*** is the smallest convex polygon containing all of them

Goal: output the vertices of the hull in counterclockwise order

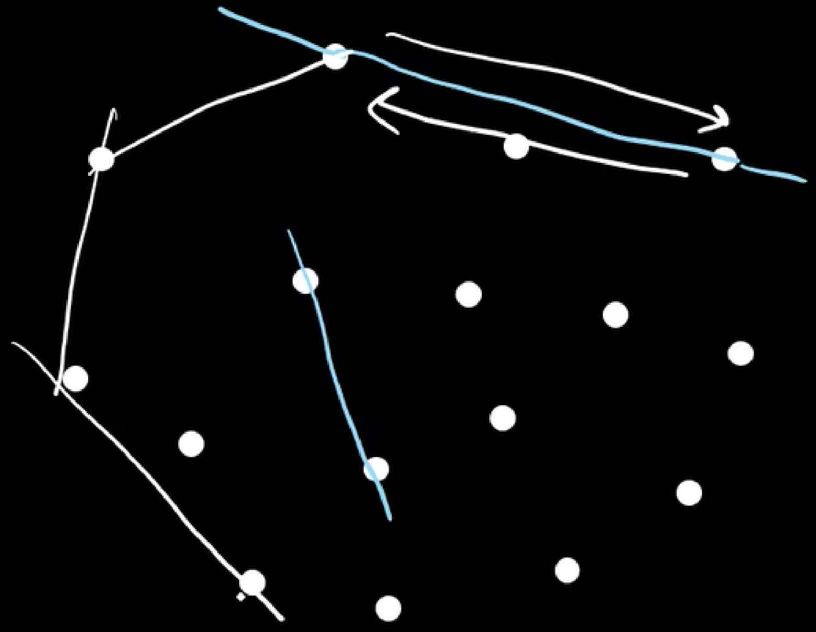


An $O(n^3)$-time algorithm

Observation (Hull edges): The edges of the convex hull must be pairs of points from the input

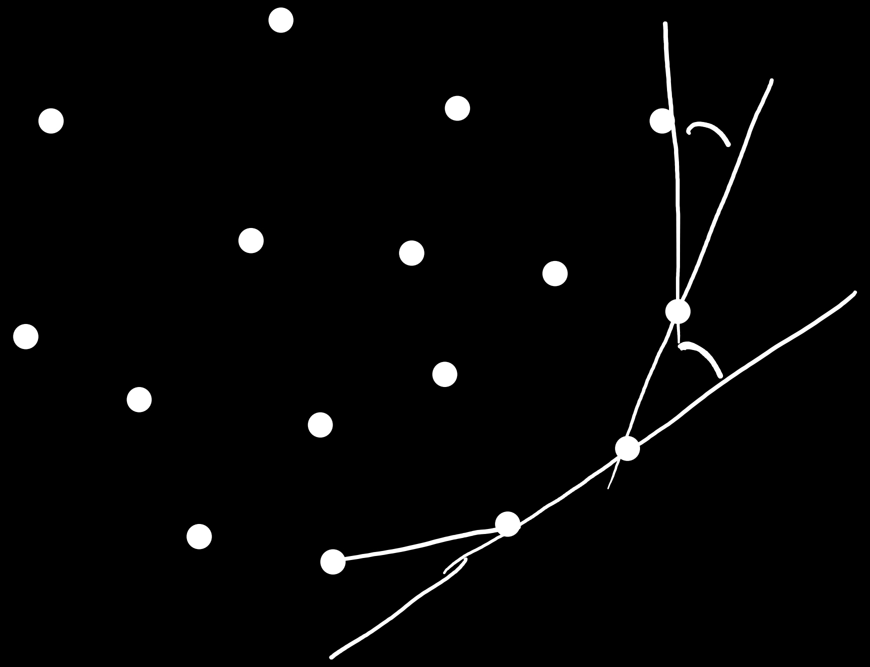
Claim (Hull edges): A segment (p_i, p_j) is on the convex hull if and only if…

all other points are on the left



Better: An $O(n^2)$-time algorithm

Observation (Order helps): The $O(n^3)$-time algorithm found the hull edges in an arbitrary order… What if we try to find them in CCW order

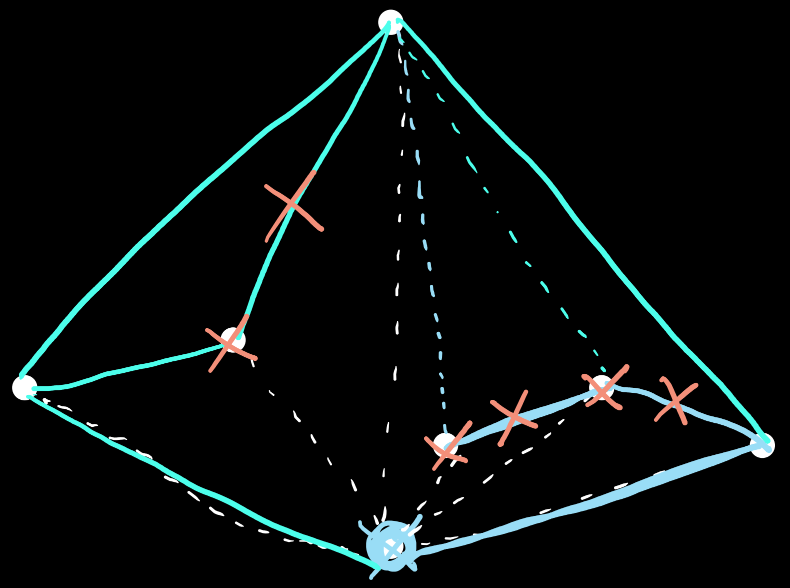


Find lowest point

Find next point with least angle

Graham Scan: An $O(n \log n)$ algorithm

Observation (Order helps again): We went from $O(n^3)$ to $O(n^2)$ by finding the edges in order… but we still processed the points in an arbitrary order. Can we order the points and do better?



Graham Scan: An $O(n \log n)$ algorithm

Algorithm:

Graham Scan

Find lowest point p_0

Sort points $p_1, p_2, \ldots$ **counterclockwise by their angle with $\boldsymbol{p_0}$**

$\boldsymbol{H = [p_0, p_1]}$
for each point $i = 2 \ldots n - 1$

    while LINESIDETEST(H[-2], H[-1], p_i) == RIGHT

        H.pop()

    H.append(p_i)

return H

Graham Scan: Complexity

Theorem: Graham Scan runs in $O(n \log n)$ time

Proof: Sorting points takes $O(n \log n)$

$O(n)$ time scan

Lower Bound

Theorem: Any convex hull algorithm that uses line-side tests to find the hull requires $\Omega(n \log n)$ line-side tests (in a decision tree model)

Won't prove this