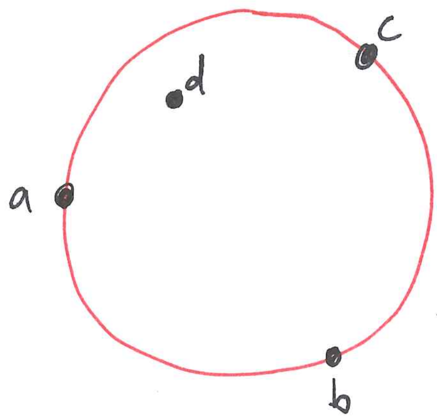
Last Time:

Linear predicates: $\text{sign}\left(\det\begin{bmatrix} a & b & c \\ 1 & 1 & 1 \end{bmatrix}\right)$

InCircle test: $$\frac{\text{sign}\left(\det\begin{bmatrix} a & b & c & d \\ \|a\|^2 & \|b\|^2 & \|c\|^2 & \|d\|^2 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right)}{\text{sign}\left(\det\begin{bmatrix} a & b & c \\ 1 & 1 & 1 \end{bmatrix}\right)}$$



normalize inside vs outside.

CAVEAT: Numerical Stability can be a problem for determinants

Key Idea

Abstract away arithmetic.

Correct implementation ~~requires~~ assumes RealRAM model. Storing real #s!

Today: Convex Hulls

Def Given $U \subseteq \mathbb{R}^d$, the convex closure of U is the set $CC(U)$ of all convex combinations of points in U.

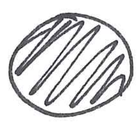
$\sum \alpha_i u_i$ (Linear), $\sum \alpha_i = 1$ (Affine), $\alpha_i \geq 0$ (Nonneg)

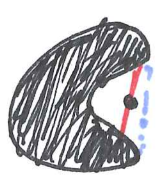
Def A set $U \subseteq \mathbb{R}^d$ is convex iff $U = CC(U)$.

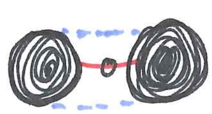
Examples:

(1) a point ✓

(2)  ✓

(3)  ✓

(4)  ✗

(5)  ✗

Facts: (1) A, B convex $\Rightarrow A \cap B$ convex

(2) $CC(U) = \bigcap_{\{V \text{ convex}:\, U \subseteq V\}} V$

Def The convex hull of U is $CH(U) = \partial CC(U)$

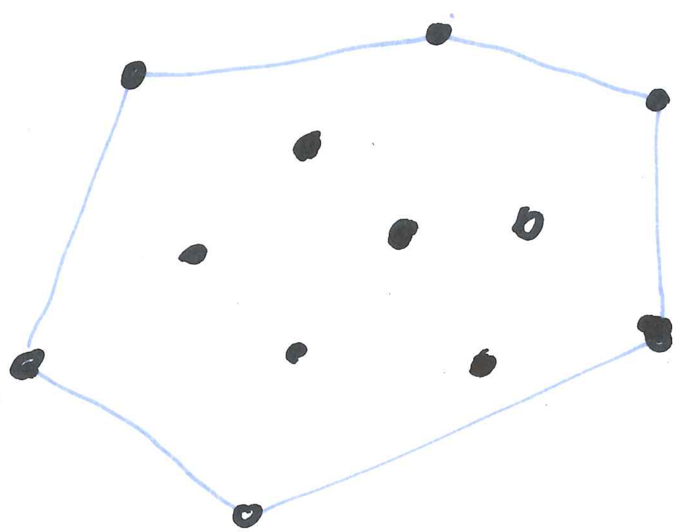
boundary

Today, we focus on convex hulls in $\mathbb{R}^2$.

INPUT: n points $P = \{p_0, \dots, p_{n-1}\}$ in $\mathbb{R}^2$
(in "general position")

no 3 collinear

OUTPUT: $CH(P)$ given as ccw ordering of vertices.



Easy for our eyes

Easy for a piece of string.

Why CH?

- Summary of points
- Outlier detection

Let's Discover an Algorithm

Find one point
Leftmost point must be in the CH.

Find the next point
Find the point that makes the biggest angle.

Keep going...
Repeat until we get back to the first point

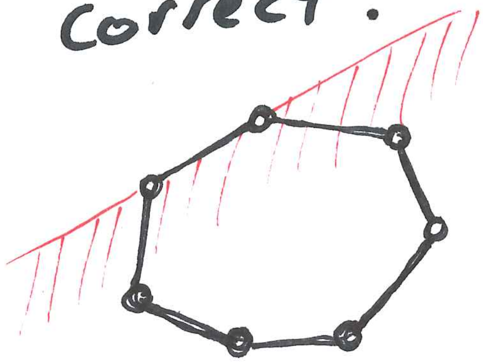
Looks Like Selection Sort...

2 ideas
(1) Use tricks from Sorting (i.e. Divide + Conquer)

(2) Use sorting directly.

Both work. Today, we'll do (2).

First, why was our naive algorithm correct?



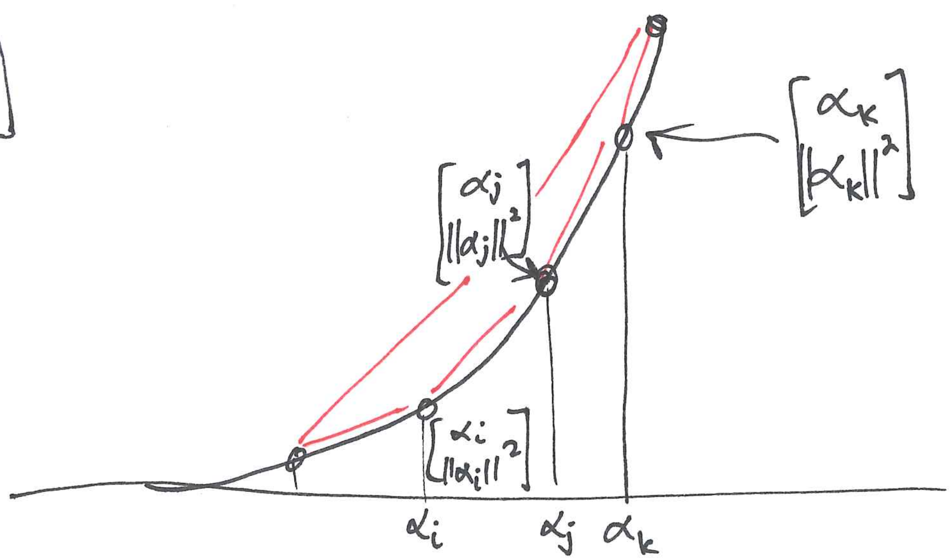
Each edge in the output has a supporting halfplane.

The CC is the intersection of these halfplanes.

A Sorting Lower Bound

Claim: Given $\alpha_1, \dots, \alpha_n \in \mathbb{R}$, we can sort $\alpha_1, \dots, \alpha_n$ in linear time given $CH\left(\begin{bmatrix}\alpha_1 \\ \|\alpha_1\|^2\end{bmatrix}, \dots, \begin{bmatrix}\alpha_n \\ \|\alpha_n\|^2\end{bmatrix}\right)$.

Proof By Picture

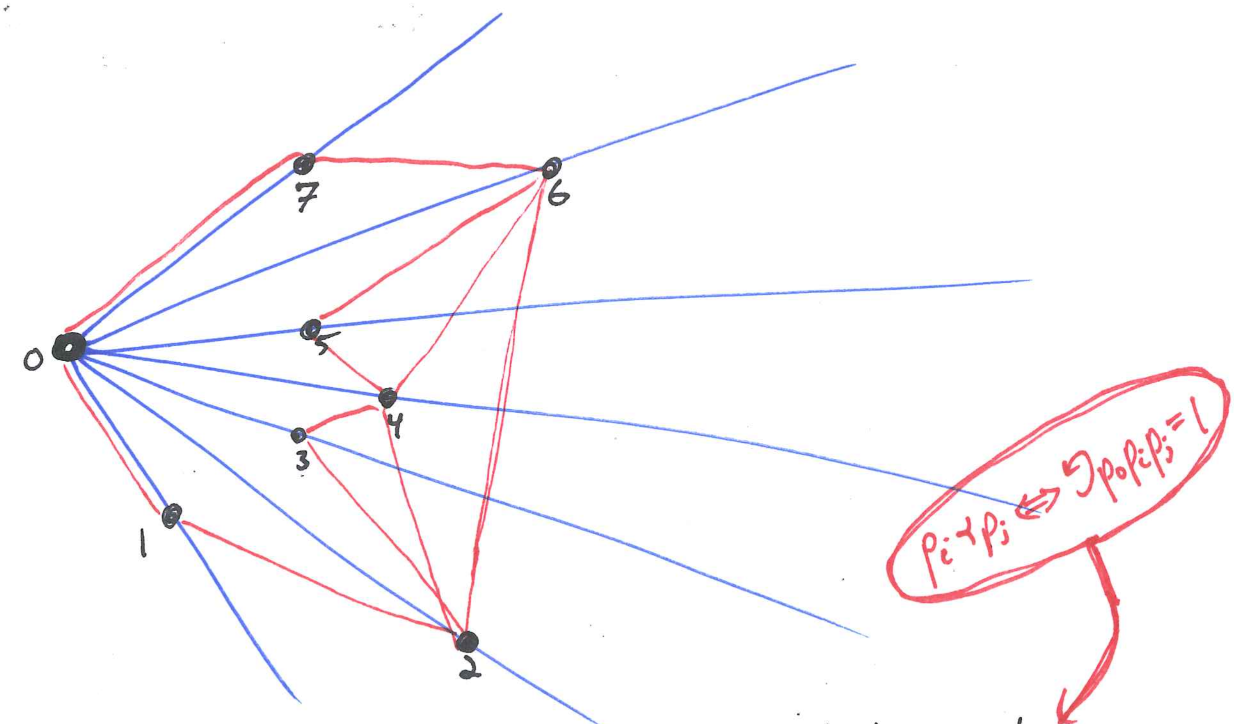


This implies $\Omega(n \log n)$ LB for CH (in comparison model).

Graham Scan: $O(n \log n)$ CH algorithm.

1972

- Sort points by angle (use linear predicates)
- Process one at a time
- Keep CH of first i pts inductively.



Let leftmost point be p_0, sort other pts by angle

Push 0, 1 to output stack

stack[0] = top element
stack[1] = 2nd element.

For $i = 2$ to $n-1$

```
{  while ( ↺(stack(1), stack(0), p_i) = -1 )
        { stack.pop }
   stack.push(p_i)
}
```

assumes general position

Analysis of Graham Scan

Naïve Analysis: Process n points. Each point can take $O(n)$ time $\Rightarrow O(n^2)$

(p_i can take $O(i)$ time)

Aggregate Analysis (a simple form of amortized analysis)

Our challenge: count stack operations.

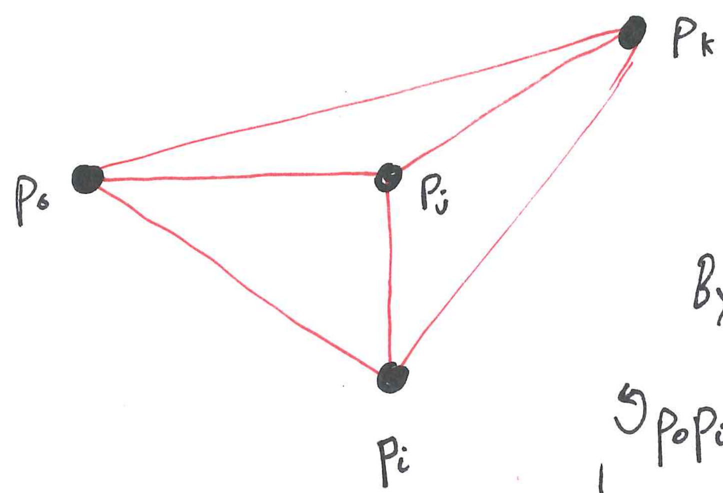
Claim: Graham Scan only requires $O(n)$ stack operations.

Why? Each point gets pushed and popped at most once each.

Each peek at top 2 elements leads to a push or a pop.

Correctness of Graham Scan

(1). All vertices of CH(P) are on the stack.



We pop p_j if $\circlearrowleft p_i p_j p_k = -1$.

By sorting, we know

$$\underbrace{\circlearrowleft p_0 p_i p_j = \circlearrowleft p_0 p_i p_k = \circlearrowleft p_0 p_j p_k}$$

$$\Downarrow$$

$$p_j \in \Delta p_0 p_i p_k \Rightarrow p_j \notin CH(P).$$

(2) $\circlearrowleft p_i p_{i+1} p_{i+2} = 1 \quad \forall i$ in output.

(all Left turns)

This is the explicit invariant maintained by the algorithm.

Lem Given $p_0, \ldots, p_{n-1}$ s.t. ($=p_n$)

(1) $\circlearrowleft p_{i-1} p_i p_{i+1} = 1 \quad \forall i = 0, \ldots, n-1$, and

(2) $\circlearrowleft p_0 p_i p_j = 1 \quad \forall i < j$

All left turns

sorted by angle from p_0

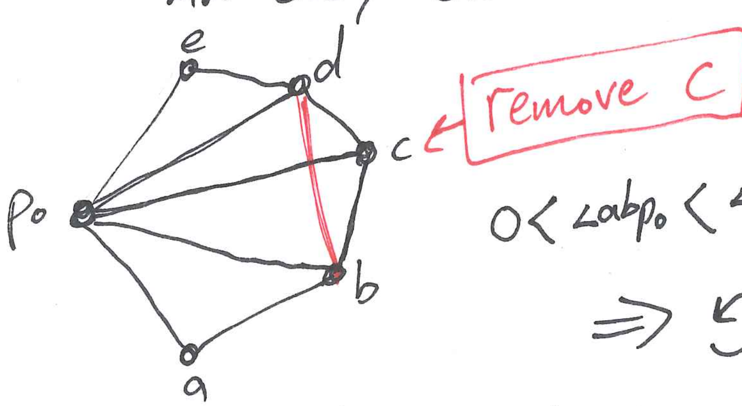
Then $p_0, \ldots, p_{n-1}$ are the vertices of a convex n-gon listed in ccw order.

---

pf/ Suffices to prove each edge $p_i p_{i+1}$ defines a support line, i.e. $\circlearrowleft p_i p_{i+1} p_j = 1 \quad \forall j \neq i, i+1$

Proof idea: Use induction. Show that removing a point other than p_i, p_{i+1}, p_j, or p_0 leaves the lem hypothesis satisfied.

Base case: $n=3$ or $n=4$.
An easy exercise.



$0 < \angle abp_0 < \angle abd < \angle abc < \pi$

$\Rightarrow \circlearrowleft abd = 1.$

$\circlearrowleft bde = 1$ by symmetric arguments.

Summary

Convex Hull of Planar Point Sets

- as hard a sorting
- Graham Scan
  - (sort w/ Lin. Pred.'s)
  - Aggregate Analysis
- Proving a polygon is convex.