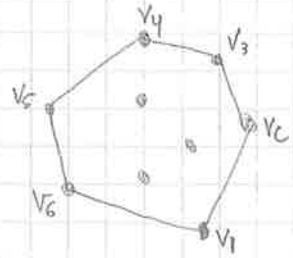
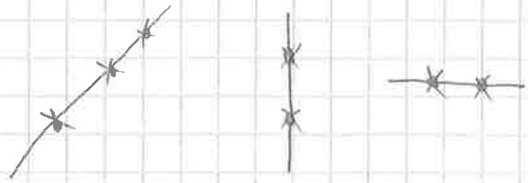
Today : Convex hull algorithms



Goal: Given a finite number of points in $\mathbb{R}^2$, determine their convex hull

For simplicity, we assume that the points are in general position:

- no 3 points on a line
- no 2 points with same x- or y-coordinate.



Fact: If $P \subseteq \mathbb{R}^2$ is finite then $\mathrm{conv}(P)$ is a convex polygon determined by its sequence $(v_1, \dots, v_n)$ of vertices ordered in clockwise/counter-clockwise order (cw/ccw)

i.e. $\mathrm{conv}(P)$ is bounded by the sequence of segments

$\overline{v_1v_2}, \overline{v_2v_3}, \dots, \overline{v_{n-1}v_n}, \overline{v_nv_1}$

Desired algorithm:

Input: $P \subseteq \mathbb{R}^2$ finite

Output: List $(v_1, \dots, v_n)$ of vertices of $\mathrm{conv}(P)$ in cw/ccw order.

(A) Naive Algorithm

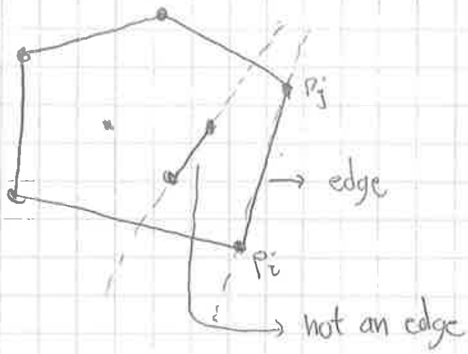
For all pairs $p_i, p_j \in P$

$O(n^3)$ ← Check whether all other points $P \setminus \{p_i, p_j\}$ are on the same side of the line $p_i p_j$

$\binom{n}{2}$ pairs, $n-2$ points to be checked

If yes ⟹ $p_i p_j$ is a convex hull edge

$O(n^2)$ ← At the end, sort all convex hull edges



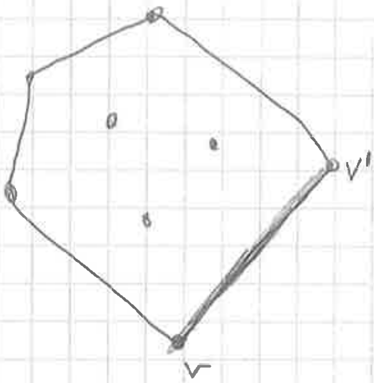
Complexity: $O(n^3)$

(B) Slight improvement

$O(n)$ ← (1) The bottommost point v of P is a convex hull vertex.

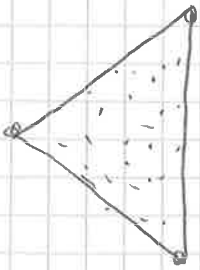
$O(n^2)$ ← (2) Look for a convex hull edge vv' as before.

n times ← (3) Look for another convex hull edge $v'v''$, and so on, until reaching v again



If h = # edges of conv(P), then

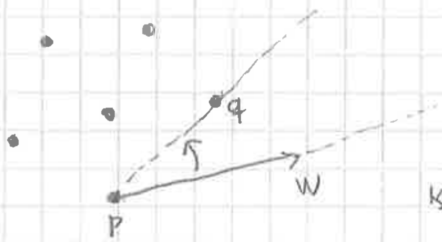
Complexity: $O(n^2 \cdot h)$ $h=n$ in worst case



$h=3$. n big

(c) Gift Wrapping Algorithm (Jarvis March), 1973

For $p \in P$ and direction w, rotate the ray from p in direction w in ccw order, until touching a first point $q \in P$



q is called the pivot of (p, w)

Key: q can be found in linear time $O(n)$ (computing angles)

$O(n) \leftarrow$ (1) Start with bottommost point v_1.

$O(n) \leftarrow$ (2) Find pivot v_2 of $(v_1, \binom{1}{0})$

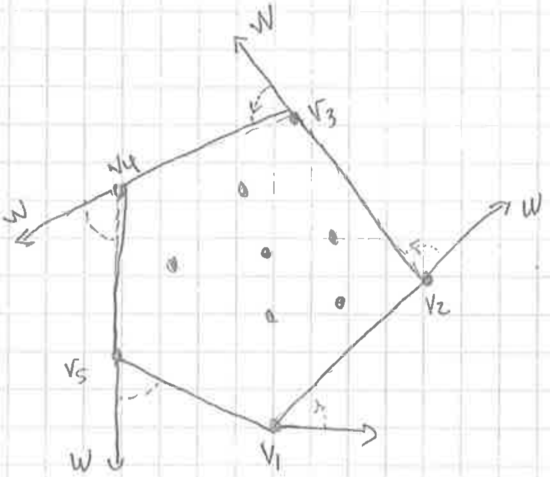
Set $w \leftarrow \overrightarrow{v_1 v_2}$, $i \leftarrow 2$

h times. (3) While $v_i \neq v_1$

$v_{i+1} \leftarrow \text{pivot}(v_i, w)$

$w \leftarrow \overrightarrow{v_i v_{i+1}}$

$i \leftarrow i+1$



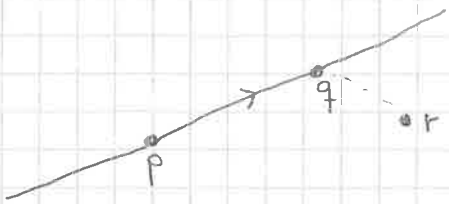
Complexity: $O(n \cdot h)$, $h \doteq$ # edges/vertices

worst case $O(n^2)$

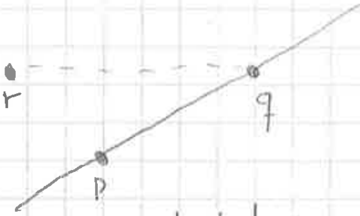
(D) Graham's Scan Algorithm, 1972

For $p, q, r \in \mathbb{R}^2$ not colinear, we say that

pqr is a <u>right turn</u> if r is on the right of the directed line $\overrightarrow{pq}$



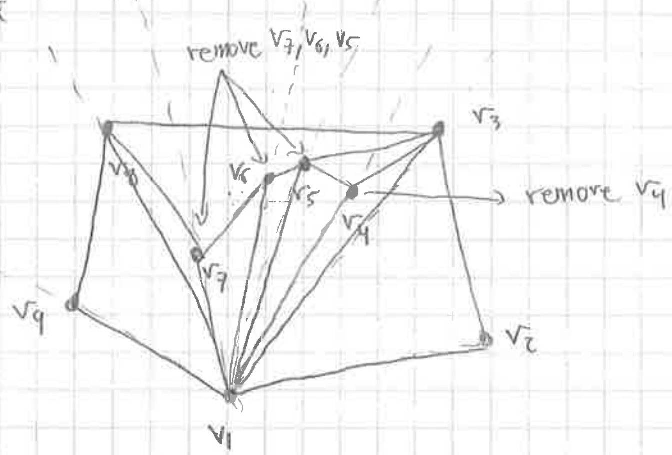
right-turn



left-turn

(1) Let v_1 be the bottommost point of P

$O(n\log n) \leftarrow$ (2) Sort $P \setminus \{v_1\}$ by angle with respect to $(v_1, \binom{1}{0})$



(3) Set $CH \leftarrow (v_1, v_2, v_3)$

(4) For $i = 4, \dots, n$

$O(n) \leftarrow$ - Append v_i to CH

- While CH ends with $(\dots, x, y, z)$ and xyz is a right-turn remove y from CH

(5) Return CH

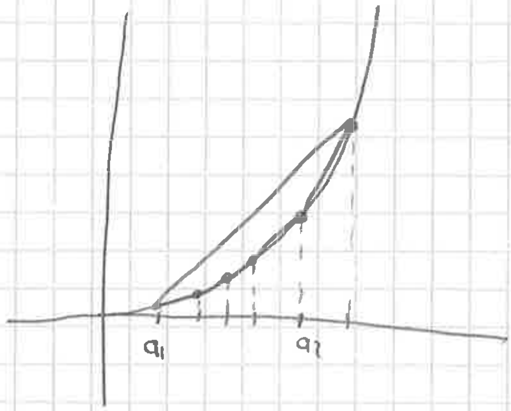
<u>Complexity</u>: $O(\underbrace{n\log n}_{\text{sorting}} + \underbrace{n}_{\text{\# of insertions to CH}} + \underbrace{n}_{\text{\# of removals from CH}}) = O(n\log n)$

Can we do better?

Lower bound

Reduce the problem of computing convex hull to sort numbers $a_1, \dots, a_n$

Let $P := \{ (a_i, a_i^2) : i = 1, \dots, n \}$



Computing $\mathrm{conv}(P)$ yields an order of the vertices

$\Rightarrow$ sorted sequence

Sorting is known to require $n \log n$ time

$\Rightarrow$ computing $\mathrm{conv}(P)$ is $\Omega(n \log n)$.