www.cs.uu.nl/geobook/

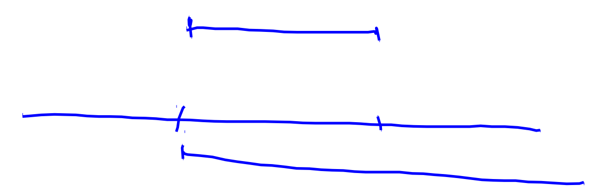
e-learning Czech version with the English pseudocodes

Convex hull in the plane

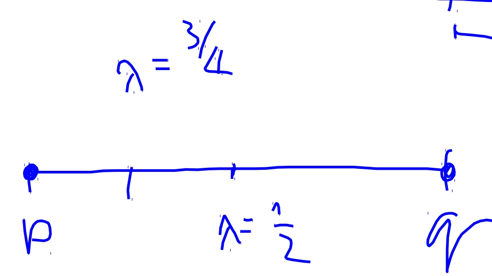
$K \subseteq \mathbb{R}^2$ convex $\forall p, q \in K$ segment $pq \subseteq K$

convex combination

$r \in pq$ $\quad r = \lambda p + (1-\lambda) q$



$\lambda \in [0,1]$



$\lambda = 1$ $\quad \lambda = \frac{1}{2}$

$\lambda = 0$

$r = (r_x, r_y)$

$r_x = \lambda p_x + (1-\lambda) q_x$

$r_y = \lambda p_y + (1-\lambda) q_y$

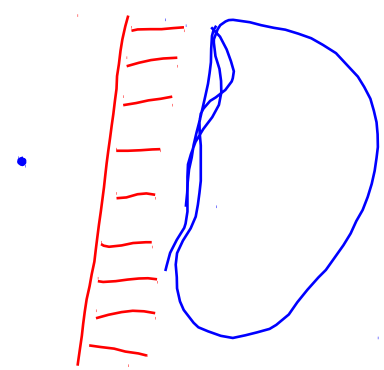
Convex hull of a set P

$CH(P)$ is the smallest convex set containing P.

$$CH(P) = \bigcap_{\substack{K \supseteq P \\ K \text{ convex}}} K$$

Plane $\mathbb{R}^2$

- biggest convex set $\mathbb{R}^2$
- any other convex set is a subset of a halfplane



P bounded

$$CH(P) = \bigcap_{\substack{H \supseteq P \\ H \text{ halfplane}}} H$$

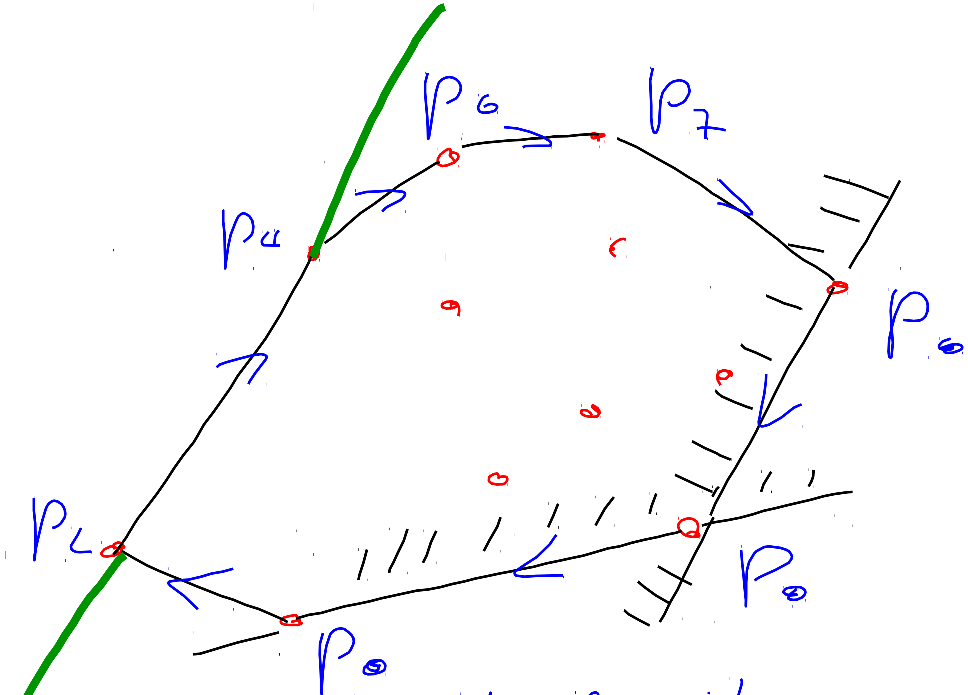
P finite

$$CH(P) = \bigcap_{H \supseteq P} H$$

boundary of H is $p_i p_j$

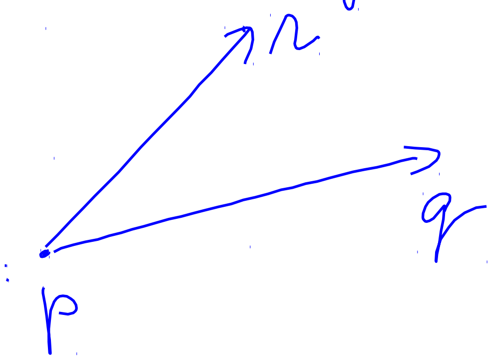
$p_i, p_j \in P$

$i \neq j$

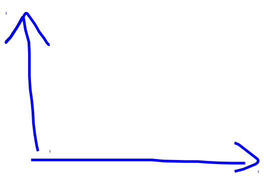


Input: A finite set P

Output: A convex hull of P given by vertices of the polygon in clockwise direction r is lying to the left of $\overrightarrow{pq}$ iff



$$\det \begin{pmatrix} q_x - p_x & r_x - p_x \\ q_y - p_y & r_y - p_y \end{pmatrix} > 0$$



> 0

$$\det \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix} > 0$$

Running time P has n points

number of different pairs of points is $n(n-1) = O(n^2)$

number of operations with one pair is $O(n)$

whole running time is $O(n^3)$

Better algorithm running time $O(n \log n)$

Second disadvantage

Unstable

Notation with O

Running time of an alg. $T(n)$ is $O(f(n))$

f is a function $\mathbb{N} \longrightarrow \mathbb{R}$

It means that there is a constant K such that for all n

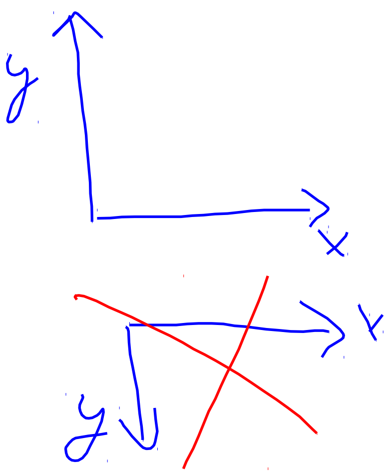
$$T(n) \leq K \cdot f(n)$$

Better algorithm – Graham's scan

Lexicographical arrangement of points in the plane.

p, q points

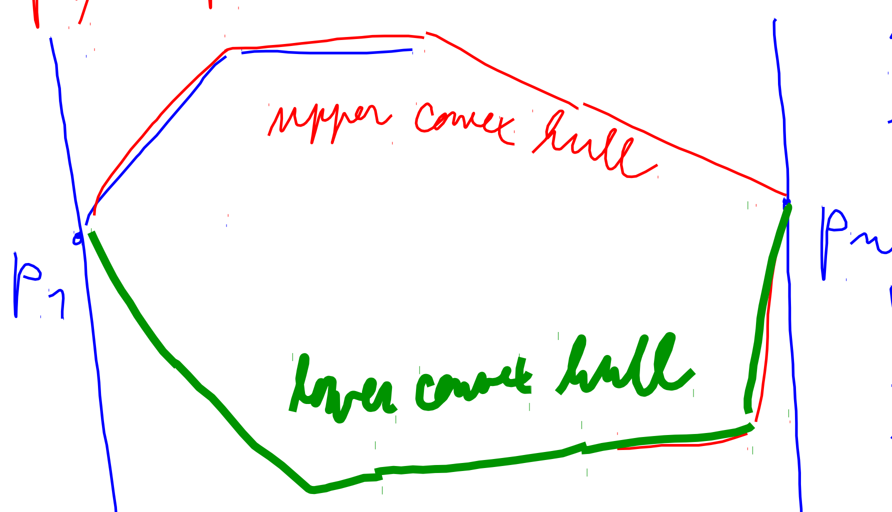
$$p < q \iff (p_x < q_x) \text{ or } (p_x = q_x \wedge p_y < q_y)$$



P a finite set with n points

we order the set lexicographically

$$p_1 < p_2 < \dots \dots < p_n$$



boundary of the convex hull can be divided into two parts

We will look for upper and lower convex hulls.

Idea: $\mathcal{L}_i$ is the upper convex hull for the set

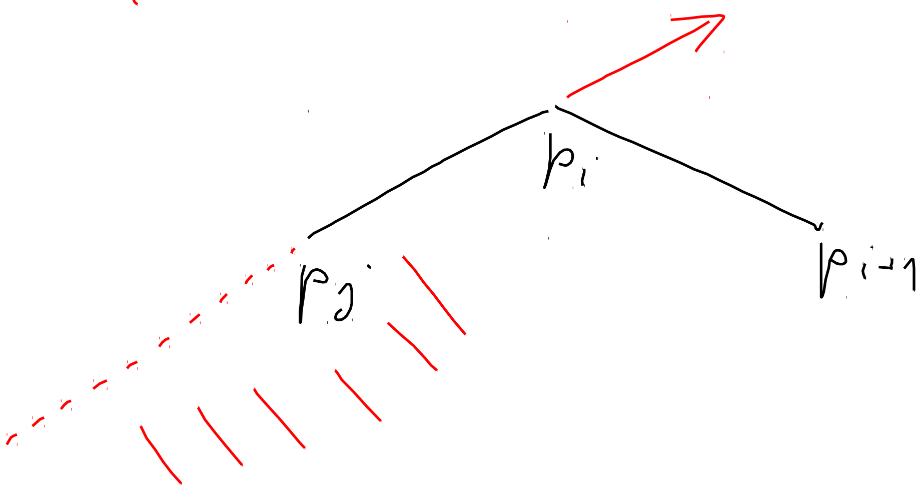
$$P_i = \{p_1, p_2, \dots p_i\}$$

$P_2 = \{p_1, p_2\}$ $\quad \mathcal{L}_2 = (p_1, p_2)$

Suppose we have $\mathcal{L}_i$ and let us make $\mathcal{L}_{i+1}$ from it.

We add point p_{i+1}.

(1) $\mathcal{L}_i = (\dots , p_j, p_i)$



p_j, p_i, p_{i+1} form a right turn

$\det(\quad) < 0$

$$\mathcal{L}_{i+1} = (\dots p_j, p_i, p_{i+1})$$

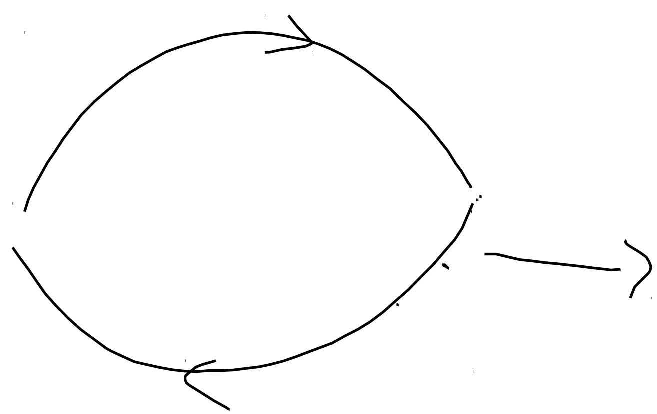
(2) p_j, p_i, p_{i+1} do not make a right turn

We add p_{i+1} to $\mathcal{L}_i$ to form $\mathcal{L}_{i+1}$.
We discard p_i (the middle point from $\mathcal{L}_{i+1}$.

Now we have to take the last three points and check if they form a right turn.

same procedure until

- there are only two points
- the last three points form a right turn



from $n-2$ to 1

<u>Theorem</u> The algorithm is correct (finds the convex hull). Its running time is $O(n \log n)$.

<u>Proof</u> Running time

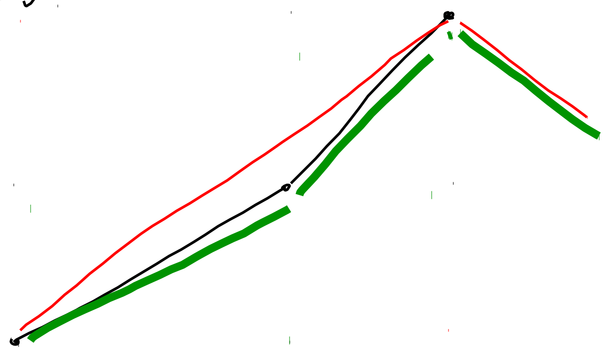
To order n points lexicographically lasts $O(n \log n)$

The rest lasts only $O(n)$ time.

- adding a point to $\mathcal{L}_{upper}$
- removing a point from $\mathcal{L}_{upper}$ at most once

$O(n)$

The algorithm is stable



$\det\left(\quad \right) \sim 0$

in fact > 0

But we can get < 0

The mistake does not interrupt the algorithm

We get convex hull with a small mistake.

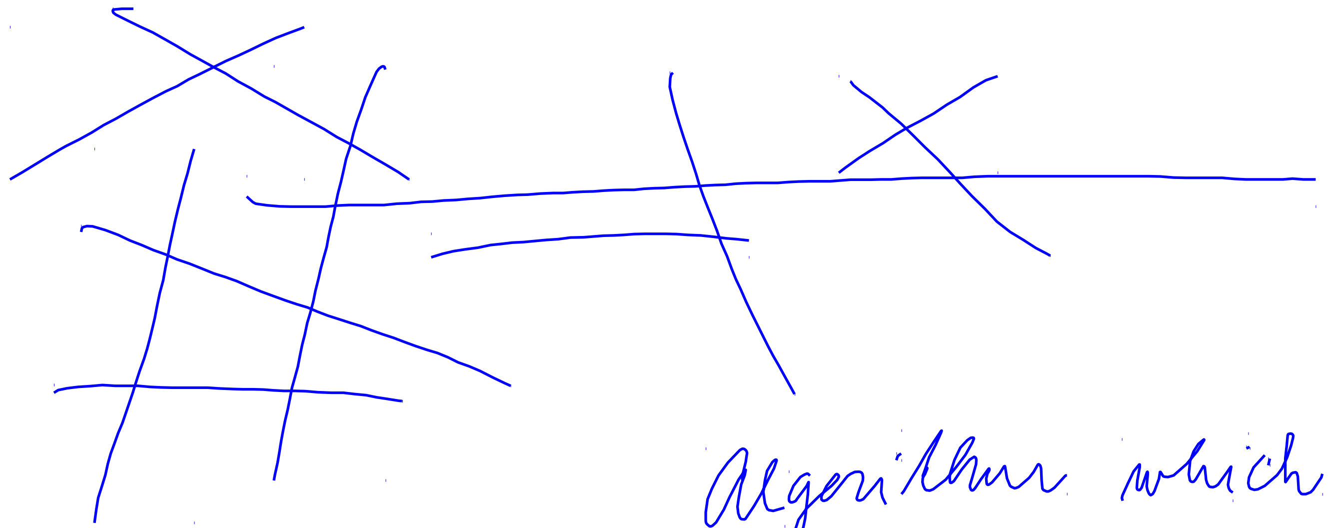
The algorithm with running time depending on the size of ~~its~~ output.

Gift wrapping algorithm

in every vertex of convex hull we spend $O(n)$ time. If there are k-vertices of the convex hull, the running time is $O(k\,n) \ll O(n \log n)$.

k small

2 Intersections of segments



n segments

$$\binom{n}{2} = \frac{n(n-1)}{2} = O(n^2)$$

Algorithm which running time depends on the size of output

Number of intersections is k

$$O\big((n+k)\log n\big)$$

Computation of intersection of two segment

$p, q \qquad pq: \; r = \lambda p + (1-\lambda) q \qquad \lambda \in [0,1]$

$s \; t \qquad st: \; o = \mu s + (1-\mu) t \qquad \mu \in [0,1]$

Intersection is given by the equation

$$\lambda p + (1-\lambda) q = \mu s + (1-\mu) t \qquad \lambda, \mu \in [0,1]$$

In coordinates

$$\begin{aligned} \lambda p_x + (1-\lambda) q_x &= \mu s_x + (1-\mu) t_x \\ \lambda p_y + (1-\lambda) q_y &= \mu s_y + (1-\mu) t_y \end{aligned} \qquad \lambda, \mu \in [0,1]$$

$$\underline{\underline{\lambda}}(p_x - q_x) + \underline{\underline{\mu}}(-s_x + t_x) = -q_x + t_x$$