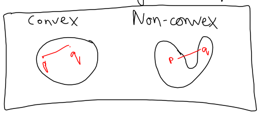
John Bourke bourkej@math.muni.cz

- Geometrical algorithms
- Each week - study an algorithm in computational geometry
- Interconnected
- Exam at end
- Elearning. Book: Computational geometry by de Berg.

zář 20-15:15

Convex hull in the plane

$K \subseteq \mathbb{R}^2$ (plane) is convex if for all $p, q \in K$ the line segment $\overline{pq}$ is in K.



$\overline{pq}$ consists of points of form $r = p + \lambda(q-p)$ for $\lambda \in [0,1]$

$\lambda = 0 \mapsto p$, $\lambda = 1 \mapsto q$.

Equivalently, $r = (1-\lambda)p + \lambda q$ or $r = \lambda p + (1-\lambda)q$.

zář 20-16:06

Convex hull of a set P:

$CH(P)$ = smallest convex set containing P

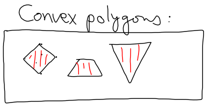
$= \bigcap_{\substack{K \text{ convex} \\ P \subseteq K}} K$

- Not computationally very useful - infinitely many convex sets containing P.
- Goal: compute $CH(P)$ for P finite.

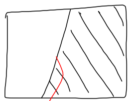
zář 20-16:11

P finite $\Rightarrow CH(P)$ convex polygon: bounded by finitely many straight line segments.

Convex polygons:



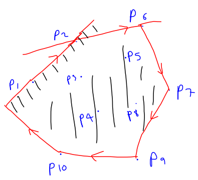
Non-convex polygon:



halfplane: points on one side of straight line

For P finite, $CH(P) = \bigcap_{P \subseteq H \text{ halfplane}} H$

zář 20-16:15

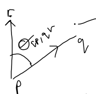


Idea: Simple algorithm

- Search for directed line segments $\vec{pq}$ on convex hull in clockwise order: eg $\vec{p_1p_2}, \vec{p_2p_6}, \vec{p_6p_7}, \ldots$
- For such a segment on $CH(P)$ no point of P will lie to its left!

zář 20-16:20

When does a point r lie to the left of $\vec{pq}$?



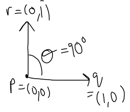
turning anticlockwise

r lies to the left of $\vec{pq}$ $\Leftrightarrow$ $0 < \theta_{qp,qr} < 180°$

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

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

eg.



$\det\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = 1$

zář 20-16:26

Algorithm: Slow Convex Hull (P)

- For each pair (p,q) test if no other points of P lie to the left of (p,q)
- If so, add $\vec{pq}$ to our list $\mathcal{L}$.
- Sort clockwise (can be done in time $O(n \log n)$) no. of inputs

Complexity: $n(n-1)$ pairs of distinct points
- For each such pair, must check $n-2$ points lie to left.

$$O(n(n-1)(n-2) + n\log n)$$
$$= O(n(n-1)(n-2)) = O(n^3).$$

zář 20-16:34

Complexity of algorithms:

Algorithm T determines function $T: \mathbb{N}_+ \longrightarrow \mathbb{R}_+$ (no of inputs n; time)

$T(n)$ = max. time it takes to compute algorithm given n inputs.

- Given $f,g: \mathbb{N}_+ \longrightarrow \mathbb{R}_+$ we write $f(x) = O(g(x))$ if $\exists\, c \in \mathbb{R}_+$ & $N \in \mathbb{N}_+$ such that $\forall n > N$ $f(n) \leqslant c\,g(n)$.

Say alg T has complexity $O(g(n))$ if $T(n) = O(g(n))$.

- $O(n)$, $O(\log n)$, $O(n^\alpha)$.. $O(n^2+3n+2) = O(n^2)$.

zář 20-16:41

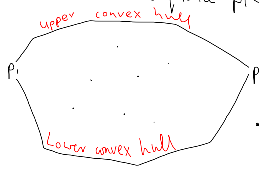
Faster algorithm: Graham's scan $O(n \log n)$

Order points in P lexicographically:

$p < q \iff p_x < q_x$ or $(p_x = q_x$ & $p_y < q_y)$

(Left to right, bottom to top)

- Obtain ordered sequence $p_1 < p_2 < \dots < p_n$.



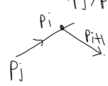
- p_1, p_n lie on the convex hull
- Convex hull splits in two parts: upper hull & lower hull
- Search for upper convex hull & then the lower convex hull.

zář 20-16:50

Let $\mathcal{L}_i$ = upper convex hull for $P_i = \{p_1, p_2, \dots, p_{i+1}\}$

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

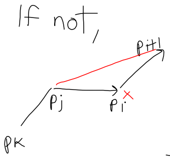
- Given $\mathcal{L}_i$ construct $\mathcal{L}_{i+1}$: add p_{i+1} which must belong to $\mathcal{L}_{i+1}$.
- Consider last 3 points in $\mathcal{L}_{i+1}$ (p_j, p_i, p_{i+1}). Say they form a right turn if p_{i+1} lies to right of directed line segment $p_j p_i$ (ie. $\det(\ ) < 0$)



- If they do form a right turn we stop.

zář 20-17:05

If not,



delete the middle point p_i from $\mathcal{L}_{i+1}$.

Then look at last 3 points in $\mathcal{L}_{i+1}$ & repeat this step until:
- last 3 points form a right turn
- only 2 pts remain in $\mathcal{L}_{i+1}$

- In this way we obtain $\mathcal{L}_{upper} = \mathcal{L}_n$.

zář 20-17:12

- Vertices of lower hull calculated in similar way: Calculate lower hull of sets $\{p_{n-i}, \dots, p_{n-1}, p_n\}$ for $i=1$ to $i=n-1$.
  ie. $\{p_{n-1}, p_n\}$, $\{p_{n-2}, p_{n-1}, p_n\}$, ... $\{p_1, \dots, p_n\}$.
- Finally, append $\mathcal{L}_{lower}$ to $\mathcal{L}_{upper}$ (first deleting p_1, p_n from $\mathcal{L}_{lower}$).

zář 20-17:19

Time complexity : $O(n \log n)$.

- Order n points lexicographically takes time $O(n \log n)$ - eg. <u>mergesort</u>.
- On upper hull, a point is added & removed at most once - at most $2n$ actions.
- On lower hull - similarly requires at most $2n$.
- Append lists - constant.

$$O(4n + n\log n) = O(n \log n)$$

zář 20-17:23

Other algorithm : Gift wrapping (Output sensitive)

Complexity $O(\underbrace{n}_{\text{number of points}} \underbrace{k}_{\text{no of vertices on convex hull}})$

- Useful if k small relative to n, since $O(kn) << O(n \log n)$.

zář 20-17:29