

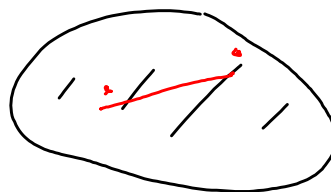
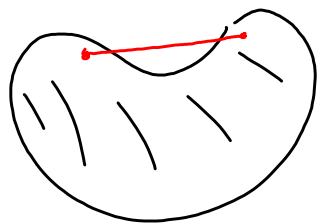
Studijni materiali

Konkretni množina (v \mathbb{R}^2 nebo \mathbb{R}^3) je taková množina K , ve které
a každými dvěma $p, q \in K$ leží i úsečka pq množina



$$x = \lambda p + (1-\lambda)q \quad \lambda \in [0,1]$$

$$p, q \in K \Rightarrow x = \lambda p + (1-\lambda)q \in K$$



Konvexni obal množiny S ($\subseteq \mathbb{R}^2, \mathbb{R}^3$) je nejmenší konvexní množina

obsahující množinu S

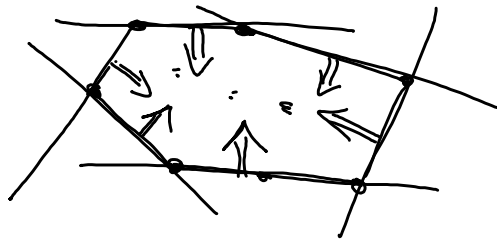
$$CH(S) = \bigcap_{\substack{K \supseteq S \\ K \text{ je konvexní}}} K$$

convex hull

Přímkou konvexních množin je konvexní množina.

Algoritmus pro konvexní obal konečných množin.

V \mathbb{R}^2 je konvexní obal konečné množiny ~~konvexní~~ mnohoúhelníkem (konvexní).



Konvexní mnohoúhelník je přímkou konvexníke počtu podporám.

Vstup: konečná množina bodů

Výstup: velmi malá množina bodů konvexní obálky rovinných n-úhelníků

$\vec{pq} \in E$
 $\vec{qr} \notin E$
 $\vec{pr} \notin E$

$\vec{p_1 p_2}$ není hranicí konvexní obálky
 8. ~~střední~~ $\vec{p_1 p_2}$ dělí body množiny S ležící npravo (mimo) $\vec{p_1 p_2}$ od

Algoritmus není robustní.

Nevýhody

- 1) ~~casova~~ náročnost
- 2) není robustní

n bodů $n(n-1)(n-2) \sim n^3$

Či má nejhorší algoritmus je $O(n^3)$.

Či má nejhorší algoritmus je $O(f(n))$, p. klížě $\left\{ \begin{array}{l} \text{určuje konstanta} \\ \text{že} \end{array} \right.$ $\&$ provedení
 algoritmu při n vstupu je $\leq K f(n)$.

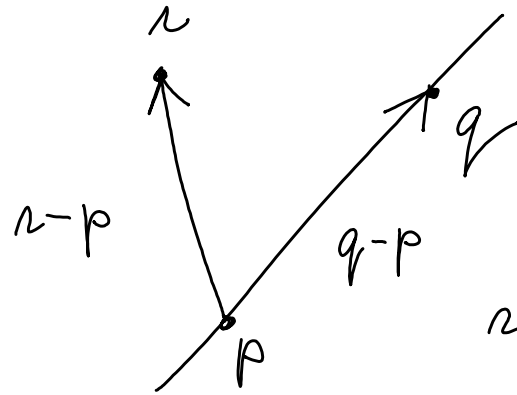
Mejistranost

• r p, q, r leží v přímce
 • q $\overrightarrow{pq}, \overrightarrow{pr}, \overrightarrow{qr}$
 • p

8. Zaleseno na kom, se každý bod
 v rychlosti m v r a p dvojnásobně na 1. místě
 a m v r dvojnásobně na 2. místě

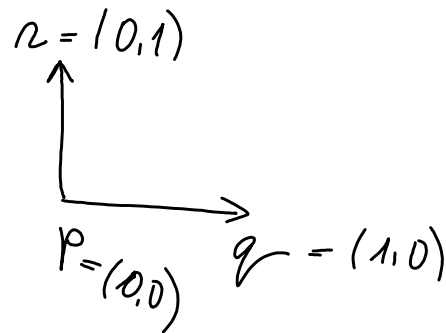
Algorithm 4.

r lies above or on the line $p \rightarrow q$:



$$\det \begin{pmatrix} q_x - p_x & r_x - p_x \\ q_y - p_y & r_y - p_y \end{pmatrix} = 0 \text{ means } r-p \text{ and } q-p \text{ are linearly dependent}$$

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



Kritérium pro to, aby bod v leží vlně od orientované přímky \vec{pq} je

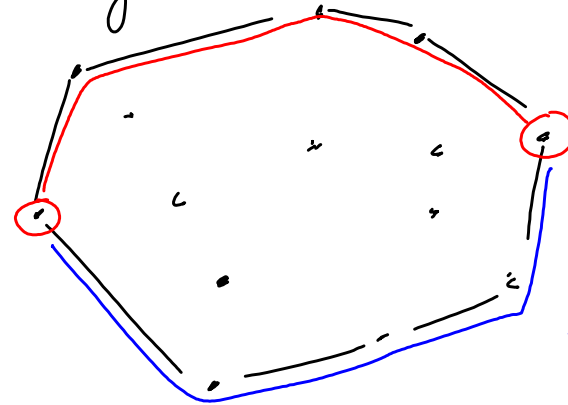
$$\det \begin{pmatrix} q_x - p_x & r_x - p_x \\ q_y - p_y & r_y - p_y \end{pmatrix} > 0$$

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

Lepsi algoritmus

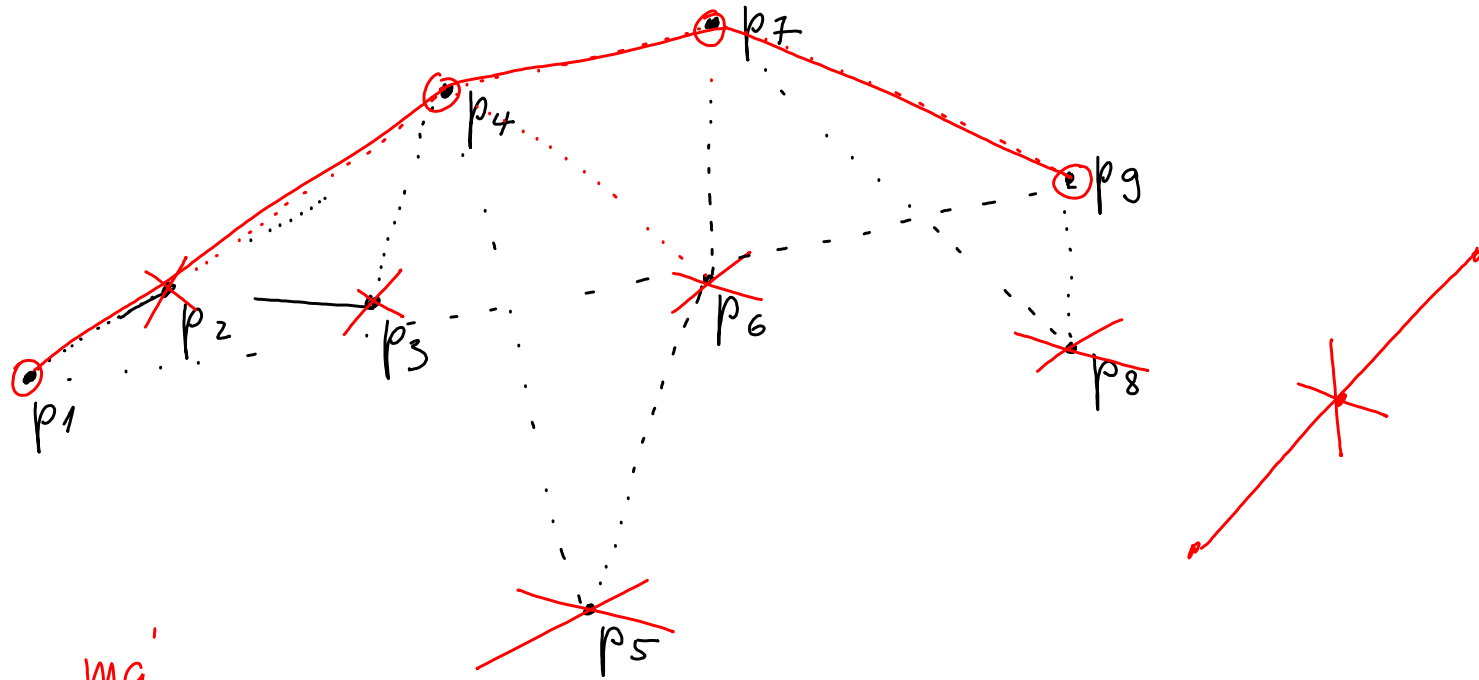
konvexní obal množiny P

Horní konvexní obal



Dolní konvexní obal

Naš algoritmus hledá horní a dolní konvexní obal
 Vtip algoritmu spočívá v tom, že si množinu bodů P
 usporádáme lexi kograficky.



Proč ^{ma'} ~~js~~ algoritmus 3.-14. časovou náročnost $O(n)$

\mathbb{R} řada \Leftrightarrow příjch algoritmu? nebo $p_x = q_x$ a $p_y < q_y$

n prvků můžeme lze uspořádat v čase $O(n \log n)$ ~~...~~

$$p_1 < p_2 < p_3 < \dots < p_n$$

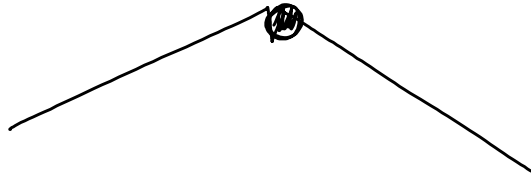
Algoritmus a řada korekce otáče po body $\{p_1, p_2, \dots, p_n\}$

myšlení řady korekce otáče po body $\{p_1, p_2, \dots, p_i, p_{i-1}\}$

Prí saohmaullosaci chytach ka algoritmus si chy, ne jaly nystede

3 body blizho sebe

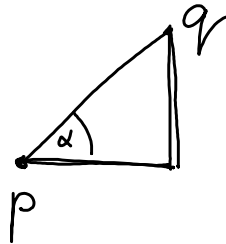
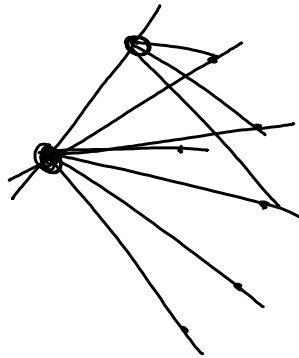
saohmaullosaci chytly nespūoli to, si algoritmus nemū iē
pohraicvat, mdau da t riak nystede



Je spoda jinych algoritmu

Gift wrapping - vhodny, pokud n me, k počet mdelu konvexniho
 abalu bude omezen ci. dem k jeho nairinost jin n bodoch
 na vstupu je $O(nk)$

Najdeme bod nejvice smero. $O(n)$



$$\tan \alpha = \frac{q_y - p_y}{q_x - p_x}$$

Najdeme bod s nejvici smernici

Vismene kento bod a poradime o nem kdez, co s podkroum, ahd
 dchud ze nedokaneme apik do bodu nejvice plero.

Toto poradime $k \times$, kde k je počet mcholu konverziko dalu.

Dalni dgeribuz ... vedik a panuj

$O(n \log n)$



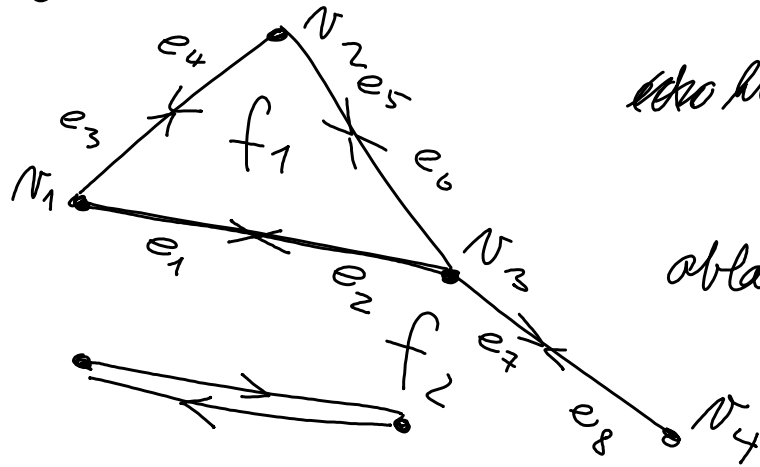
Prékryvy map

Zadávaní map ... rovinné podrozdělení

Tři typy objektů

ncholy

obkoly (edge) ... polokraný
ovembrané kraný

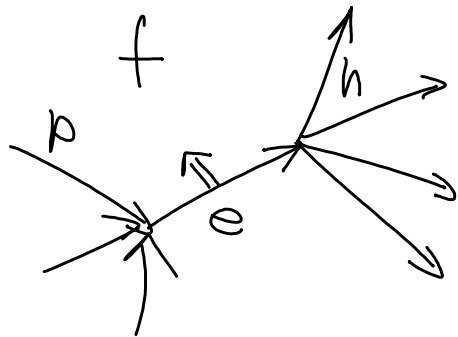


oblaste - souvislé části roviny
mezi hranami

Popis podrobněji pomocí dvojité souvislé roviny
(double connected edge list)

Taluka po mchly množina vnitřní a vnější hrana
peintler

Taluka po hrany množina vnitřní hrana (dvojice)
peintler



následující hrana množina 2 dvanáct hrany a n
přilehlá oblast stejnou přilehlou oblast
předchozí hrana (předchůdce)