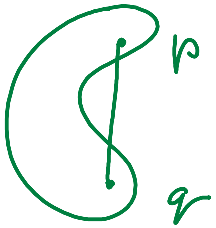
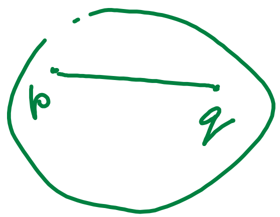
Konvexní obal v rovině

$M \subseteq \mathbb{R}^2$ je konvexní, jestliže s každými dvěma body p a q obsahuje i úsečku, která je spojuje.





$$p = (p_x, p_y)$$

$$q = (q_x, q_y)$$

Bod úsečky $\quad r = \lambda p + (1-\lambda) q \qquad \lambda \in [0,1]$

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

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

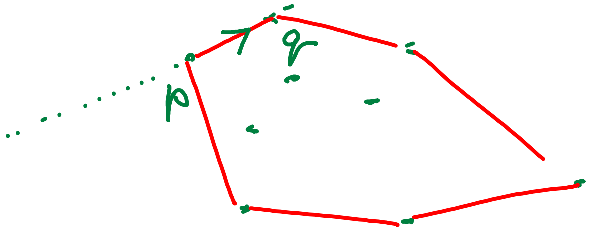
2

M libovolná množina v $\mathbb{R}^2$, konvexní obal množiny M značíme

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

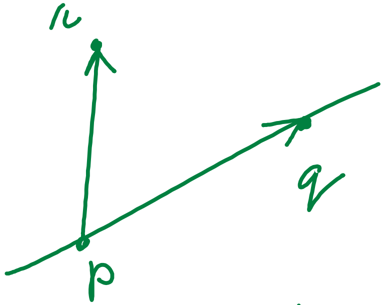
= nejmenší konvexní množina obsahující množinu M.

Je-li M konečná množina $\{p_1, p_2, \dots p_n\}$, pak jejím konvexním obalem je konvexní mnohoúhelník (průnik konečného počtu polorovin určených nějakými dvojicemi bodů z M.)



3

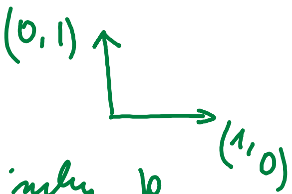
Bod r leží vlevo od orientované přímky pq



$q - p$

$r - p$

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



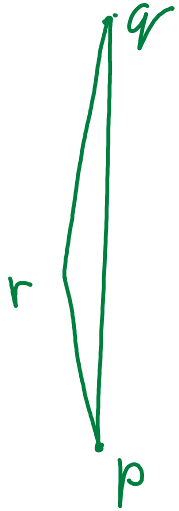
Podmínka nutná, aby r leželo vlevo od orient. přímky pq je

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

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

5

Dalši problem rešiva v kom. če algoritmus vrne: odločitev



$$pq \in E$$

$$pr \; ? \; E$$

$$rq \notin E$$

6

Lepší algoritmus

Horní a dolní konvexní obal

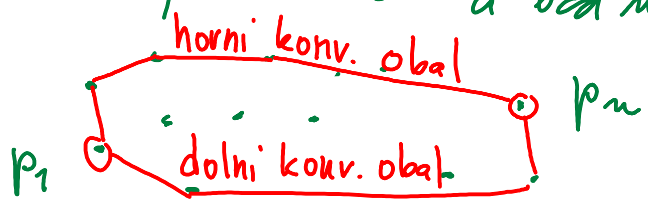
$P = \{p_1, \dots, p_n\}$ množina bodů v rovině

Lexikografické uspořádání v rovině

$p < q$ jestliže $p_x < q_x$

nebo $p_x = q_x$ a $p_y < q_y$

Najdeme bod nejvíce vlevo a bod nejvíce vpravo



p_1 a p_2 leží na hranici konvexního obalu.

7

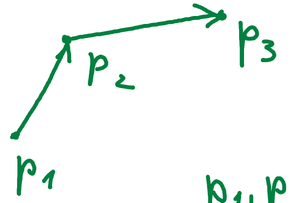
Podstatným rysem algoritmu je, že body $p_1, p_2 \ldots p_n$ uspořádáme v lexikografickém uspořádání.

Předpokládejme

$$p_1 < p_2 < \ldots < p_n$$

Algoritmus vytváří horní a pak dolní konvexní obal U

$p_1, p_2 \in U$



p_3 leží vpravo od orient. přímky $p_1 p_2$

Říkáme, že p_1, p_2, p_3 dělají „zatáčku vpravo"

p_1, p_2, p_3 vložíme v U

8

Jestliže p_1, p_2, p_3 nedělají zatáčku vpravo, pak vypustíme p_2 z $\mathcal{U}$

Obecný krok: Máme $p_1, \dots, p_s \in \mathcal{U}$ kde všechny indexy jsou $\le i-1$.

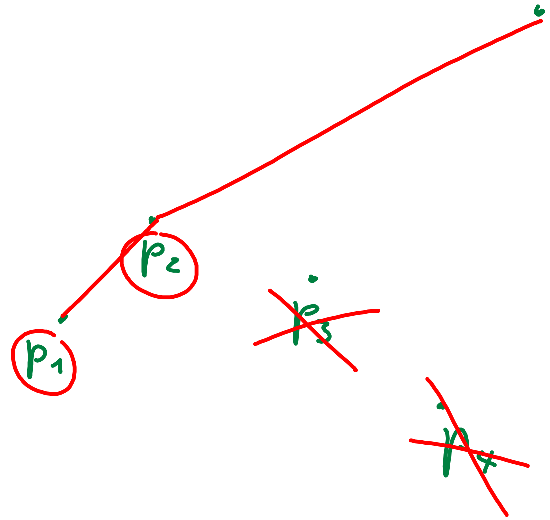
Přidáme do $\mathcal{U}$ bod p_i.

(1) poslední tři body v $\mathcal{U}$ dělají zatáčku vpravo
 => pokračujeme přidáním p_{i+1}, pokud $i \le n-1$.

(2) poslední 3 body nedělají zatáčku vpravo
 Prostřední z nich vyndáme z $\mathcal{U}$ a znovu testujeme poslední 3 body v $\mathcal{U}$.

(9)

Toto provádíme tak dlouho, dokud nenastane (1) nebo dokud U neobsahuje pouze 2 body.



Po přidání p_5
- vypustíme p_4
- vypustíme p_3

Obdobně nalezneme dolní konvexní obal. U něj začneme bodem p_n a přidáváme postupně $p_{n-1}, p_{n-2}, \ldots p_1$.

(10)

Časová náročnost algoritmu

(1) Uspořádání $p_1, p_2, \ldots, p_n$ lexikograficky

$$O(n \log n)$$

(2) Každý přidáme do U pouze jedenkrát a z U jej vypustíme nejvýše jedenkrát k tomu provedeme pouze konečný počet kroků

Tedy zbývající část algoritmu má časovou náročnost

$$O(n)$$

Celková časová náročnost je $O(n \log n)$

(11)

Symbolika $O(f(n))$

Říkáme, že časová náročnost algoritmu o vstupech velikosti n je $O(f(n))$ kde $f: \mathbb{N} \to \mathbb{R}$ nějaká funkce,

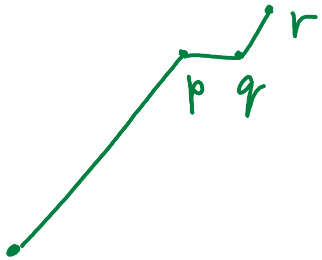
jestliže existuje konstanta C taková, že při vstupech velikosti n proběhne algoritmus v čase $\leq C \cdot f(n)$

$$\exists C > 0 \; \forall n \in \mathbb{N} \quad T(n) \leq C f(n)$$

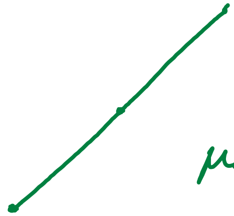
12

Algoritmus je robustní

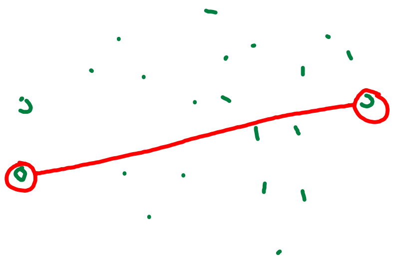
3 body velice blízko sebe



výsledek
dostaneme
vždy



prostřední bod
nedělá potíže



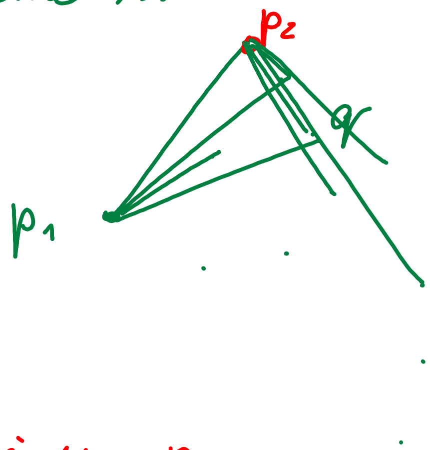
13

V situaci, kdy očekáváme, že konvexní obal bude „malý",
je výhodnější použít jiný algoritmus, jehož časová náročnost
je závislá na velikosti výstupu.
Algoritmus GIFT WRAPPING má časovou náročnost
$O(n \cdot k)$
kde n je počet bodů množiny P, jejíž konv. obal
hledáme a k je počet vrcholů konv obalu

(14)

<u>Příbalový algoritmus</u>

Začneme bodem množiny t. nejníže vlevo $(O(n))$.



počítáme

$$tg\,\alpha = \frac{q_y - p_{1y}}{q_x - p_{1x}}$$

najdeme q s největším α

$q = p_2$

Teď z na p_2

$p_1, p_2, p_3 \ldots p_k\ p_1$

Časová náročnost $O(nk)$

k-krát projdeme $O(n)$ bodů