

## 1. KONVEXNÍ OBAL V ROVINĚ

**Konvexní množiny a konvexní obaly v rovině.** Připomeňme si, že množina  $K$  v rovině je konvexní, jestliže s každými dvěma body obsahuje i úsečku, kterou určují.

OBR 1.1 Konvexní a nekonvexní množina

Každý bod  $r$  na úsečce  $pq$  lze zapsat jako tzv. konvexní kombinaci bodů  $p$  a  $q$ :

$$r = \lambda p + (1 - \lambda)q, \quad \text{kde } \lambda \in [0, 1].$$

Pro souřadnice  $(r_x, r_y)$  bodu  $r$  to znamená

$$\begin{aligned} r_x &= \lambda p_x + (1 - \lambda)q_x, \\ r_y &= \lambda p_y + (1 - \lambda)q_y. \end{aligned}$$

Průnik konvexních množin je evidentně opět konvexní množina. Konvexní obal  $\mathcal{CH}(P)$  množiny  $P$  v rovině je nejmenší konvexní množina obsahující množinu  $P$ , množinově to lze zapsat takto:

$$\mathcal{CH}(P) = \bigcap_{K \supseteq P \text{ konvexní}} K.$$

My se budeme zabývat algoritmy, které efektivně najdou konvexní obal pro konečnou množinu  $P$ . V tomto případě bude

$$\mathcal{CH}(P) = \bigcap_{H \supseteq P \text{ polorovina}} H.$$

Přitom se stačí omezit na poloroviny určené dvojicí bodů z množiny  $P$ , v nichž leží i zbývající body množiny  $P$ . Konvexním obalem konečné neprázdné množiny s aspoň dvěma body je tedy konvexní mnohoúhelník, který může zdegenerovat na úsečku.

**Naivní algoritmus.** V této sekci si ukážeme, že pouhá znalost matematické definice ještě nestačí k vytvoření dobrého algoritmu. Algoritmus bude hledat orientované úsečky  $pq$ , které tvoří hranici konvexního obalu orientovanou ve směru hodinových ručiček. Je založen na tom, že v tomto případě neleží nalevo od orientované přímky  $pq$  žádný bod množiny  $P$ .

OBR 1.2 Nalevo od orientované přímky  $pq$  neleží žádný bod množiny  $P$ .

Skutečnost, že bod  $r$  leží nalevo od orientované přímky  $pq$  lze matematicky zachytit podmínkou na determinant, jehož řádky tvoří souřadnice vektorů  $q - p$  a  $r - p$ :

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

Tento determinant matice  $2 \times 2$  lze zapsat i jako determinant matice  $3 \times 3$

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

1

## ALGORITMUS 1 z pseudo.pdf

Časová náročnost tohoto algoritmu při  $n$  bodech na vstupu je  $O(n^3)$ , neboť procházíme všechny orientované přímky a pro každou testujeme, kde leží zbývající body. Dalším nedostatkem je skutečnost, že algoritmus je nestabilní. Jestliže tři body  $p_1, p_2, p_3$  leží na jedné přímce, může nastat situace, kdy orientované úsečky  $p_1p_2, p_2p_3$  a  $p_1p_3$  jsou prvky množiny hraničních úseček  $E$ . Pak ale může nastat problém s jejich uspořádáním do posloupnosti ve směru hodinových ručiček.

OBR 1.3 Tři body množiny  $P$  na jedné hraniční úsečce.

**Lepší algoritmus.** Daleko lepší výsledek dostaneme, když body množiny  $P$  před hledáním hraničních úseček konvexního obalu vhodným způsobem uspořádáme. Použijeme lexikografické uspořádání bodů nejdřív podle souřadnice  $x$  a potom podle souřadnice  $y$ . Tedy

$$p < q \quad \text{právě když } p_x < q_x \text{ nebo } (p_x = q_x \text{ a } p_y < q_y).$$

V tomto uspořádání označme body od nejmenšího k největšímu

$$p_1 < p_2 < p_3 < \dots < p_{n-1} < p_n.$$

Body  $p_1$ , který je "nejvíce vlevo", a bod  $p_n$ , který je "nejvíce vpravo", tedy určitě leží na hranici konvexního obalu a rozdělují ji na horní a dolní část. Abychom se vyhnuli dlouhým formulacím, budeme poněkud nepřesně mluvit o horním a dolním konvexním obalu, místo abychom mluvili o horní a dolní části hranice konvexního obalu.

OBR 1.4 Horní (modrý) a dolní (zelený) konvexní obal.

Uspořádání bodů horní hranice konvexního obalu ve směru hodinových ručiček koresponduje s jejich lexikografickým uspořádáním. Proto náš algoritmus bude postupně hledat horní konvexní obal množin

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

od  $i = 2$  do  $i = n$ . Podobným způsobem pak bude postupně hledat dolní konvexní obal množin

$$\{p_{n-i}, p_{n-1+1}, \dots, p_{n-1}, p_n\}$$

pro  $i = 1$  do  $n - 1$ .

Nechť  $\mathcal{L}$  je lexikograficky uspořádaný seznam vrcholů horního konvexního obalu množiny  $P_i$ . Ten vždy obsahuje body  $p_1$  a  $p_i$ . Nyní změním seznam  $\mathcal{L}$  tak, aby byl tvořen vrcholy horního konvexního obalu množiny  $P_{i+1}$ . V něm musí být lexikograficky největší bod  $p_{i+1}$ , proto jej přidáme do  $\mathcal{L}$ .

Body nepatřící do horního konvexního obalu množiny  $P_{i+1}$  vyloučíme ze seznamu  $\mathcal{L}$  postupně takto: Vezmeme v  $\mathcal{L}$  poslední tři vrcholy  $p_j < p_i < p_{i+1}$ . Jestliže bod  $p_{i+1}$  leží vpravo od orientované přímky  $p_jp_i$ , budeme říkat, že body  $p_j, p_i$  a  $p_{i+1}$  tvoří *zátáčku vpravo*. V tomto případě seznam  $\mathcal{L}$  měnit nebudeme, neboť body množiny  $P_{i-1}$  leží vpravo od orientované přímky  $p_ip_{i+1}$ . Máme tedy seznam vrcholů horního konvexního

obalu množiny  $P_{i+1}$  a můžeme začít hledat seznam vrcholů horního konvexního obalu množiny  $P_{i+2}$ , je-li  $i + 2 \leq n$ .

OBR 1.5 Zatáčka vpravo.

Jestliže body  $p_j$ ,  $p_i$  a  $p_{i+1}$  netvoří zatáčku vpravo, pak ze seznamu  $\mathcal{L}$  vyloučíme prostřední z nich, tedy  $p_i$ . Jestliže totiž  $p_i$  leží na úsečce  $p_j p_{i+1}$ , tak ho k popisu horního konvexního obalu nepotřebujeme, i když na něm leží. Jestliže  $p_{i+1}$  leží vlevo od orientované přímky  $p_j p_i$ , pak  $p_i$  leží vpravo od orientované přímky  $p_j p_{i+1}$  a není tedy v horním konvexním obalu množiny  $P_{i+1}$ .

OBR 1.6 Situace, kdy poslední tři body v  $\mathcal{L}$  netvoří zatáčku vpravo.

Po vyloučení bodu  $p_i$  ze seznamu  $\mathcal{L}$  vezmeme opět poslední tři body tohoto seznamu, pokud existují, a provedeme s nimi stejnou proceduru. Tu provádíme tak dlouho, až

- v seznamu  $\mathcal{L}$  zůstanou pouze dva body,
- nebo poslední tři body seznamu  $\mathcal{L}$  tvoří zatáčku vpravo.

Tím je hledání seznamu vrcholů horního konvexního obalu pro množinu  $P_i$  ukončeno. Tento postup si ilustrujme na příkladu pomocí následující animace.

ANIMACE 1.1

Vrcholy dolního konvexního obalu získáme analogicky.

**Pseudokód a časová náročnost.** Výše uvedený slovní popis můžeme formalizovat pomocí následujícího pseudokódu:

ALGORITMUS 2 z pseudo.pdf

**Věta 1.1.** *Výše uvedený algoritmus je korektní a jeho časová náročnost v závislosti na počtu bodů množiny  $P$  je  $O(n \log n)$ .*

*Důkaz.* Důkaz, že pomocí algoritmu dostaneme seznam vrcholů konvexního obalu množiny  $P$  uspořádaný ve směru hodinových ručiček se redukuje na důkaz, že algoritmus nám dá seznam vrcholů horního, popř. dolního konvexního obalu seřazený lexikograficky. Důkaz pro horní obal se provádí induktivně: Ukážeme, že ze seznamu vrcholů pro horní konvexní obal množiny  $P_i$  vytvoří algoritmus seznam vrcholů horního konvexního obalu množiny  $P_{i+1}$ . To provádíme stejnými úvahami jako při slovním popisu algoritmu. Prakticky zaměřeného čtenáře jimi již nebudeme unavovat. Teoreticky zaměřený čtenář se je může pokusit zvládnout sám.

Co se týče časové náročnosti, tak první krok, uspořádání  $n$  bodů v lexikografickém uspořádání trvá, jak je dobře známo, čas  $O(n \log n)$ . Ukážeme, že zbývající část algoritmu zabere pouze čas  $O(n)$ . Každý z bodů množiny  $P$  do seznamu  $\mathcal{L}$  pro horní konvexní obal vkládáme právě jednou a nejvýše jednou jej vyndáváme. Tyto akce provedeme v konstantním čase. Tedy časová náročnost vytvoření seznamu musí být úměrná počtu bodů.

Čtenář jistě sám přijde na řadu způsobů, jak výše uvedený algoritmus vylepšit. Nicméně žádné z nich časovou náročnost nevylepší na nějaký odhad lepší než  $O(n \log n)$ .  $\square$

**Jiný algoritmus.** Pro konvexní obal v rovině existuje řada dalších algoritmů. N označíme jeden z nich, který je vhodný použít v případě, pokud dopředu víme, že počet vrcholů konvexního obalu bude vzhledem k počtu bodů množiny  $P$  malý. Postup algoritmu připomíná balení, proto se nazývá Gift wrapping.

Pro zjednodušení předpokládejme, že žádné tři body neleží v přímce. Nejdříve najdeme bod množiny  $P$  nejvíce vlevo (přesněji, nejmenší ve výše uvedeném lexikografickém uspořádání) a označme jej  $p_1$ . Z bodu  $p_1$  vedeme přímky do dalších bodů množiny  $P$  a počítáme jejich směrnice. Nechtě  $p_2$  je takový bod, že  $p_1p_2$  má z těchto přímek největší směrnici. Nyní z bodu  $p_2$  vedeme spojnice do zbývajících  $n - 2$  bodů množiny  $P$ . Z nich vybereme bod  $p_3$  tak, aby konvexní úhel  $\angle p_1p_2p_3$  je největší. Tímto způsobem pokračujeme, až se dostaneme zpět do bodu  $p_1$ . Tím jsme zakončili balení množiny  $P$  do konvexního obalu. Je-li konvexní obal  $k$ -úhelník, je časová náročnost tohoto algoritmu  $O(kn)$ . V každém z  $k$  vrcholů strávíme čas  $O(n)$ .

ANIMACE 1.2