



**OI-OPPA. European Social Fund  
Prague & EU: We invest in your future.**

---

# Melkmanův algoritmus konvexní obálky

Václav Strnad  
FEL - OI

# Obsah

Teorie

Příklad

Zdrojový kód

# Teorie

- Jedná se o online algoritmus
- Vypočítává obálku pro jednoduché křivky
- Vychází z množiny  $C$ , obsahující  $n$  vrcholů

$$C = (v_0, v_1, \dots, v_{n-1})$$

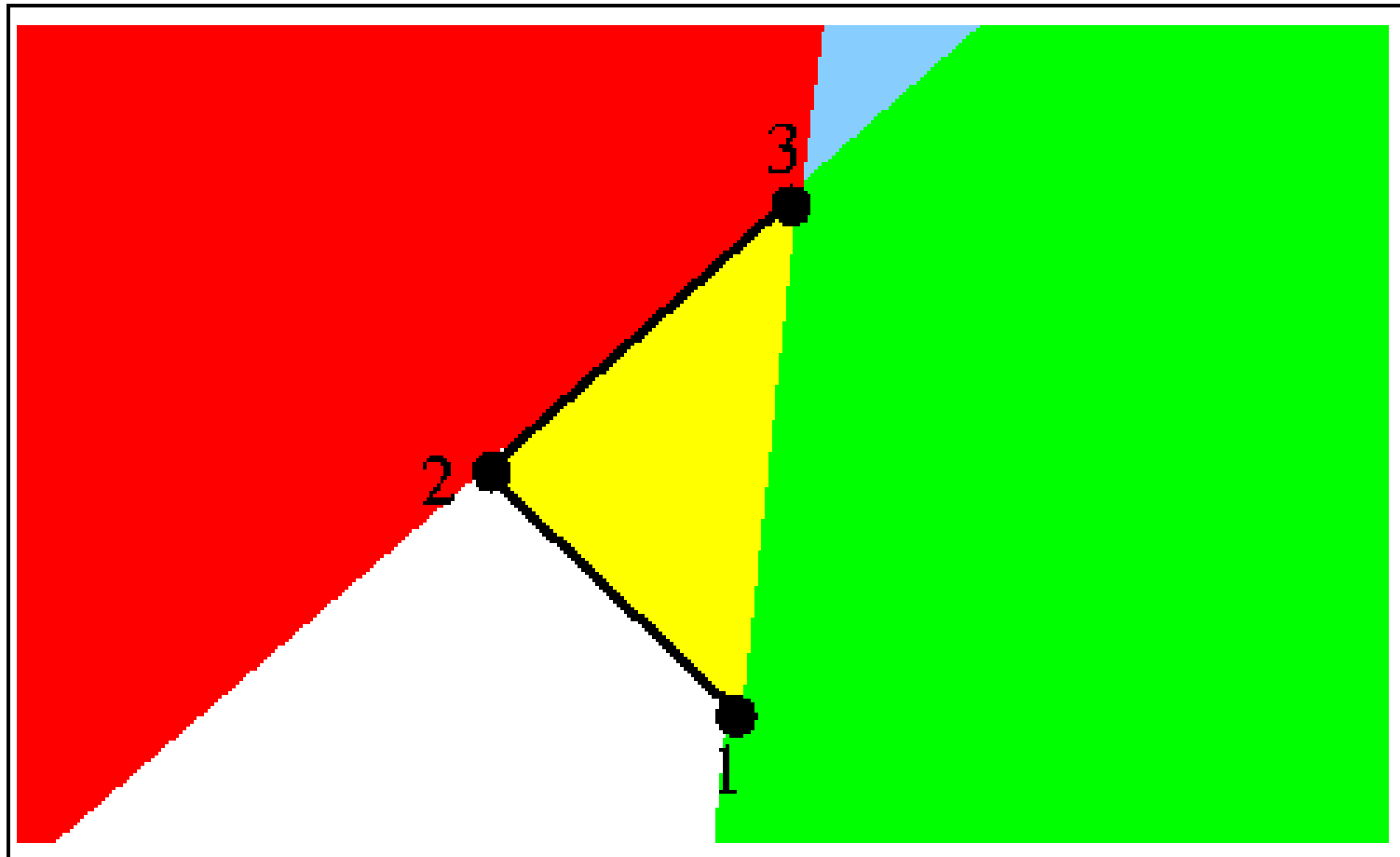
- Využívá frontu  $D$  se dvěma konci

$$D = \langle d_b, d_{b+1}, \dots, d_{t-1}, d_t \rangle$$

# Teorie – složitost algoritmu

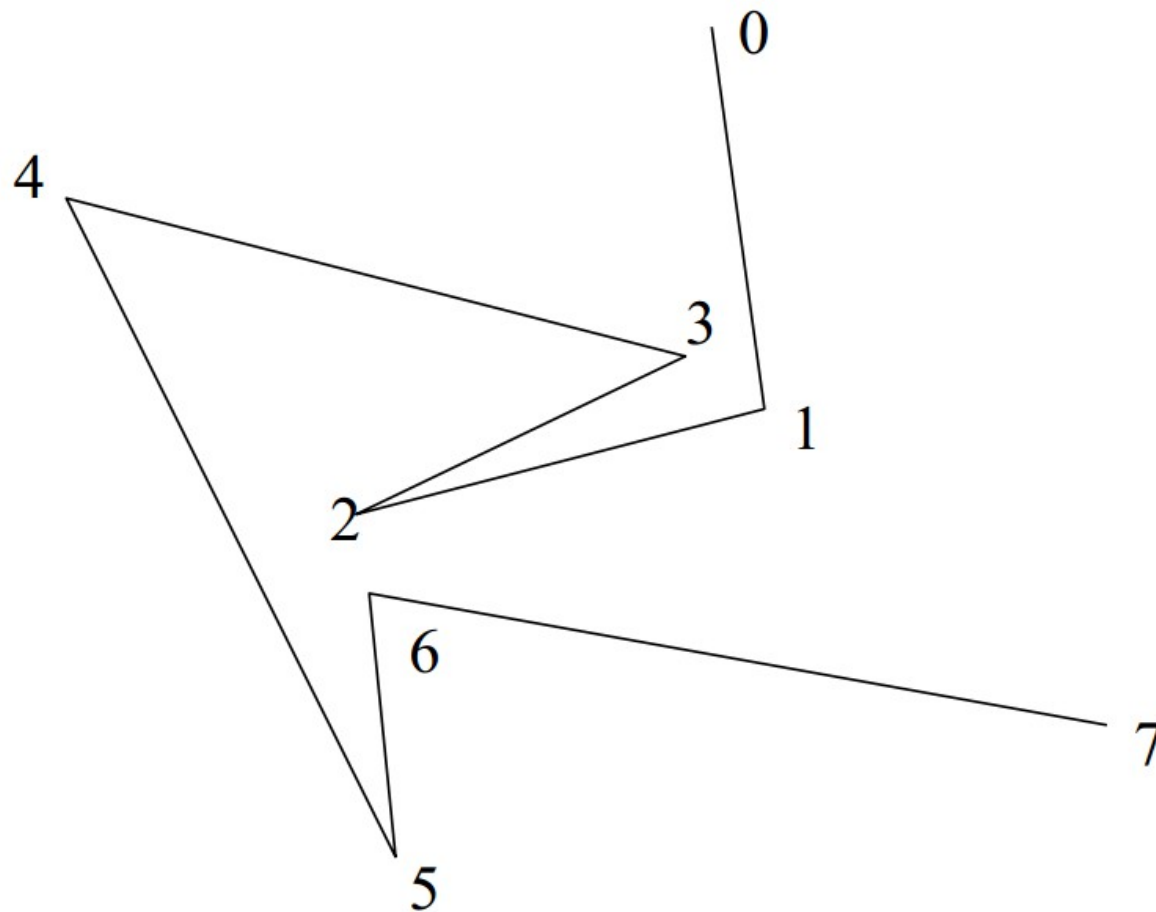
- Asymptotická výpočetní složitost:  $O(n)$
- Nejhorší:  $3n$  (maximálně 2 možnosti pro vložení a 1 pro odebrání)
- Nejlepší:  $2n$  (v případě, že první trojice je výslednou obálkou)

# Teorie – poloha nového bodu



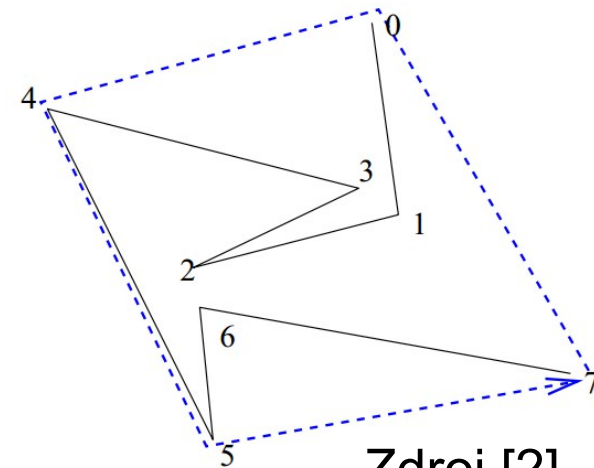
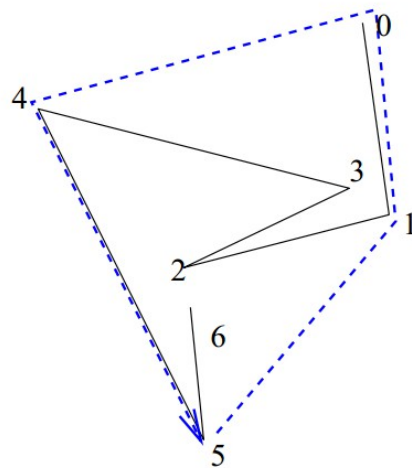
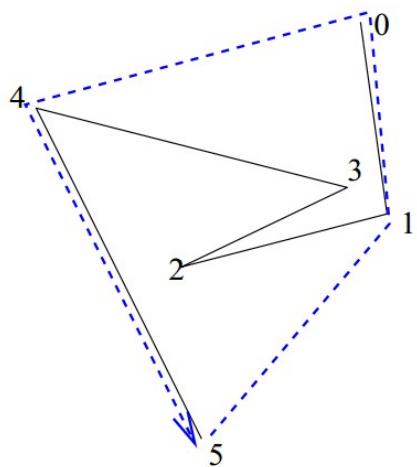
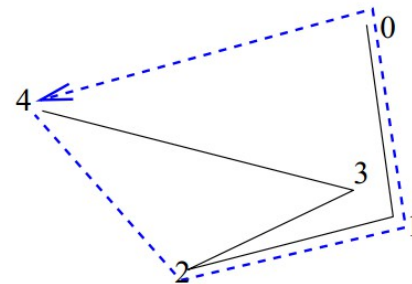
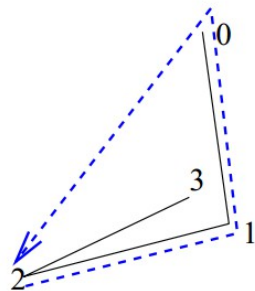
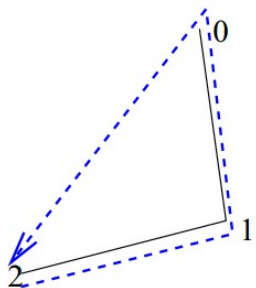
Zdroj [3]

# Příklad



Zdroj [2]

# Příklad - postup



Zdroj [2]



# Příklad - výsledek

$\langle 7; 0; 4; 5; 7 \rangle$

# Zdrojový kód – zjištění polohy bodu

- P0, P1 - body přímky, P2 - zjišťovaný bod

```
isLeft ( Point P0, Point P1, Point P2 ) {  
    return (P1.x - P0.x)*(P2.y - P0.y) - (P2.x - P0.x)*(P1.y -  
    P0.y);  
}
```

- $>0$  ... P2 je vlevo od přímky
- $=0$  ... P2 leží na přímce
- $<0$  ... P2 je vpravo od přímky

# Zdroje

- [1]SoftSurfer: Convex Hull of a 2D Simple Polyline. [online]. [cit. 2012-10-09]. Dostupné z:  
[http://softsurfer.com/Archive/algorithm\\_0203/algorithm\\_0203.htm](http://softsurfer.com/Archive/algorithm_0203/algorithm_0203.htm)
- [2]MITCHELL, Joe. [online]. [cit. 2012-10-09]. Dostupné z:  
<http://www.ams.sunysb.edu/~jsbm/courses/545/melkman.pdf>
- [3]The Computational Geometry Lab at McGill: Melkman 1987. [online]. [cit. 2012-10-09]. Dostupné z:  
<http://cgm.cs.mcgill.ca/~athens/cs601/Melkman.html>

Děkuji za pozornost.

Prostor pro otázky.



**OI-OPPA. European Social Fund  
Prague & EU: We invest in your future.**

---