

KMA/PDB

Algoritmy prostorových databází

Karel Janečka

Tvorba materiálů byla podpořena z prostředků projektu FRVŠ č. F0584/2011/F1d



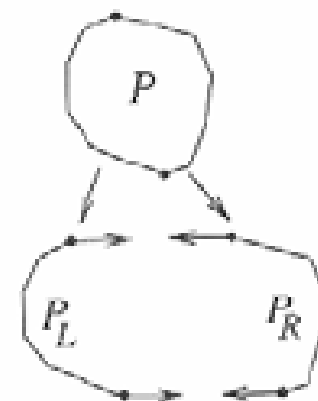
Obsah

- Průsečíky konvexních polygonů.
- Průsečíky jednoduchých polygonů.
- Lokace bodu.
- „windowing“ a „clipping“.
- Nalezení průsečíků linií.
- Konzistence prostorových databází.



Detekce průsečíku konvexních polygonů

- Mějme polygon P s L vrcholy a polygon Q s M vrcholy.
- 1) Najdeme v P a Q vrcholy s max. a min. souřadnicí y
- 2) Rozdělíme hranice polygonů na 2 polootevřené konvexní řetězce P_L, P_R, Q_L, Q_R .
- 3) P a Q se protínají $\Leftrightarrow P_L$ a Q_R se protínají a nebo P_R a Q_L se protínají.
- $O(\log n)$.

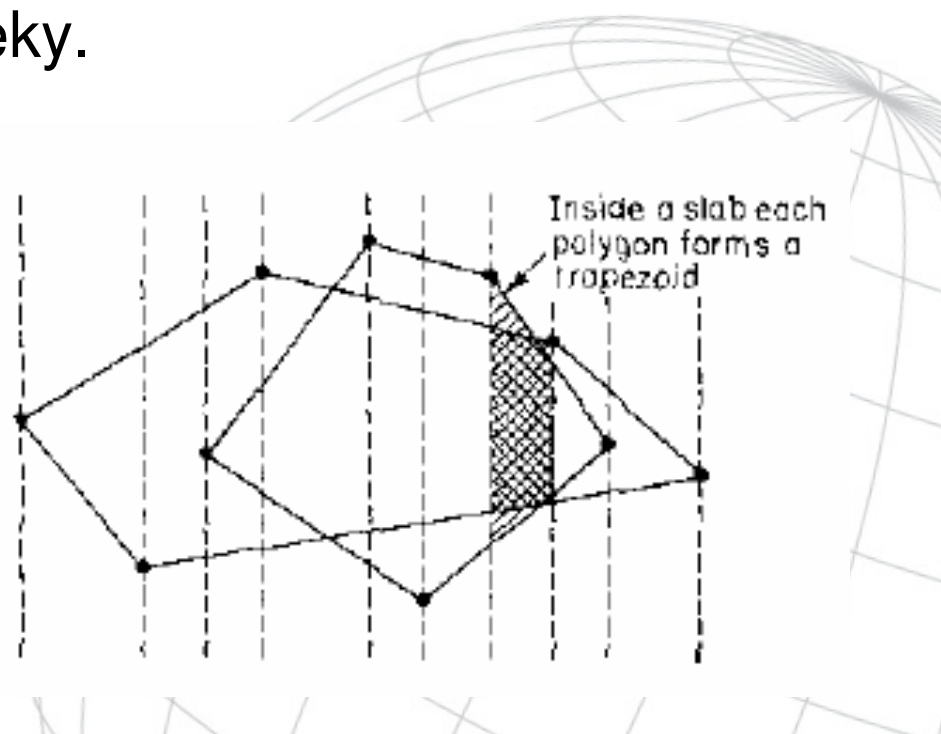


Nalezení průsečíku konvexních polygonů

- Např. algoritmus Shamosa a Hoeyho:

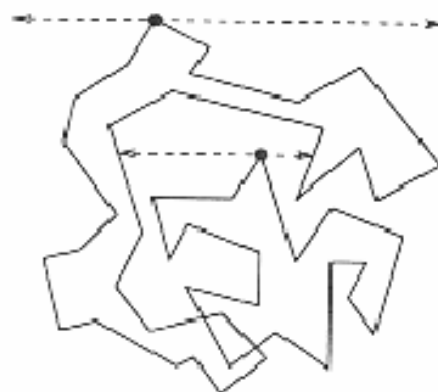
- 1) Rozdělit na pásy, ty seřadit podle x.
- 2) V pásu určit průnik dvou lichoběžníků (trojúhelníků).
- 3) Pospojovat výsledné úseky.

- $O(n)$.



Detekce průsečíku jednoduchých polygonů

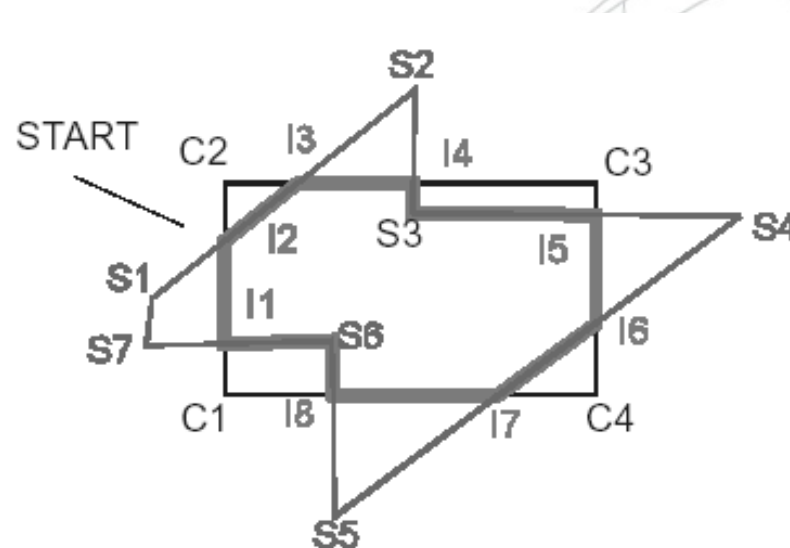
- Je polygon jednoduchý? → Test sebeprotínání:
- Předpokládáme, že ano, že polygon je jednoduchý a zkusíme triangulizovat; pokud chyba $\Rightarrow$ sebeprotíná se.
- Detekce průniku 2 jedn. polygonů: spojit polygony do jednoho uzavřeného řetězce úzkým kanálem a provést test sebeprotínání.



X / \ / \ X

Nalezení průsečíku jednoduchých polygonů

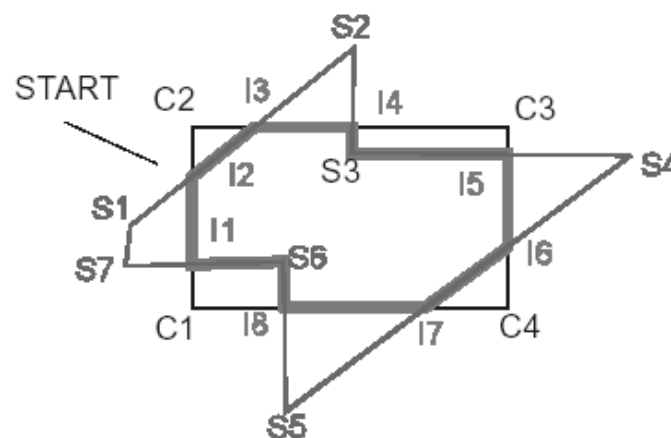
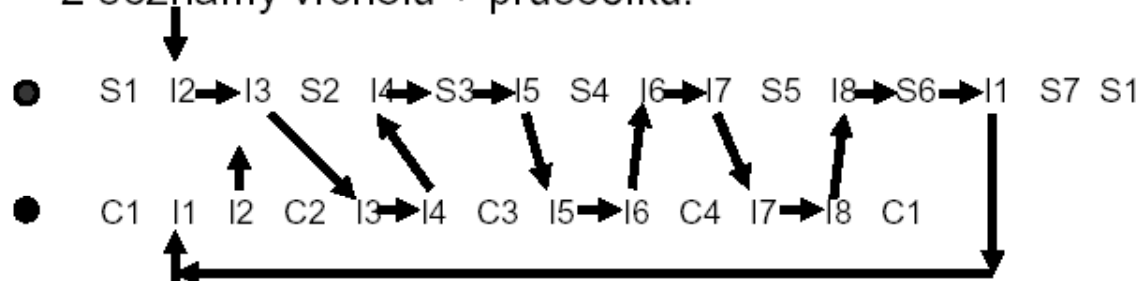
- Např. Weiler – Athertonův alg. – i pro polygony obsahující díry, tzn. nejen pro jednoduché polygony.
- Vstup: N-úhelníky zadané ve směru hodinových ručiček, díry obráceně.
- Úkolem nalézt seznam vstupních (I2, I4, I6, I8) a výstupních (I1, I3, I5, I7) průsečíků:



Nalezení průsečíku jednoduchých polygonů

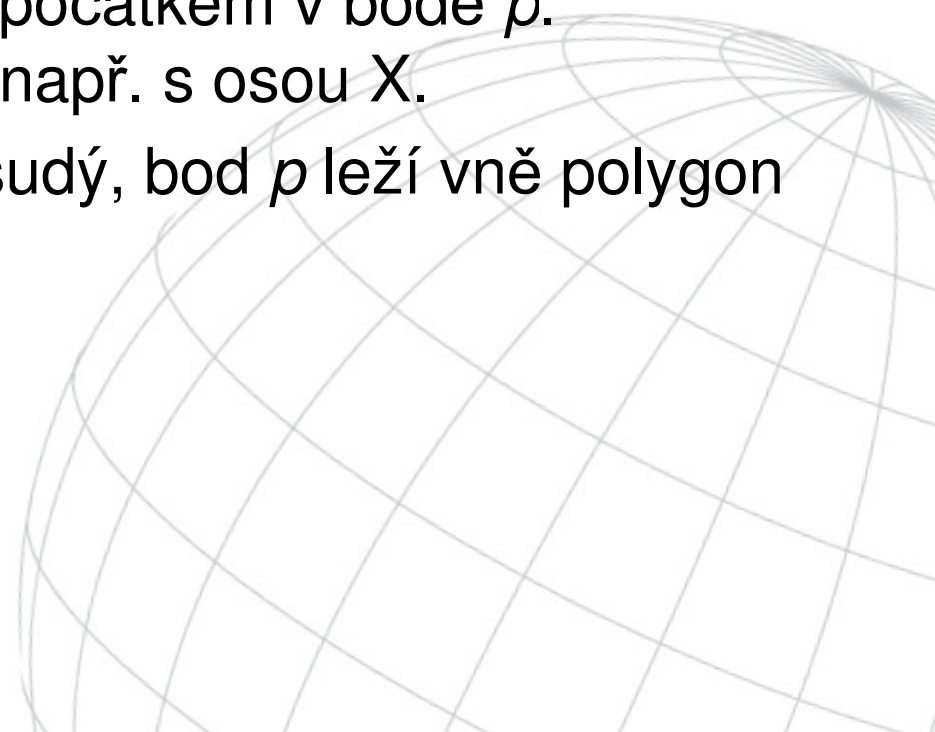
- Nyní máme dva seznamy vrcholů a průsečíků:

2 seznamy vrcholů + průsečíků:



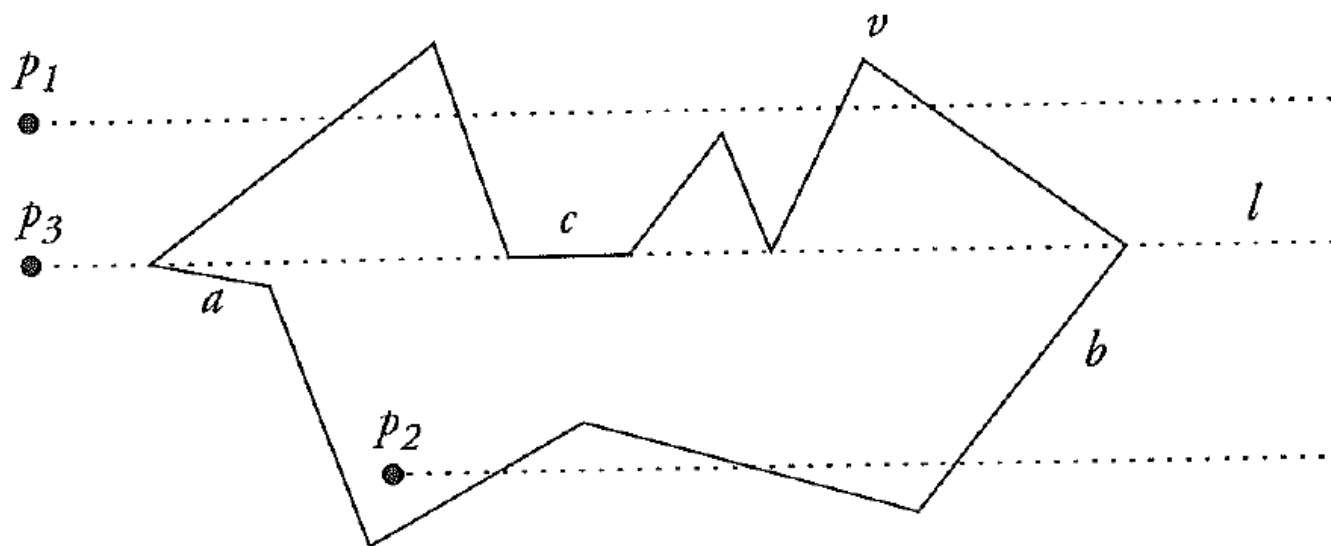
Lokace bodu

- Mějme polygon P a bod p . Úkolem je zjistit, zda se bod nachází uvnitř či vně daného polygonu.
- **Ray-crossing** (i pro nekonvexní polygon) → procházíme postupně hrany polygonu P a testujeme, zda má hrana průsečík s polopřímku l s počátkem v bodě p . Polopřímka je rovnoběžná např. s osou X .
- Pokud je počet průsečíků sudý, bod p leží vně polygon P , pokud lichý, leží uvnitř.
- $O(n)$.



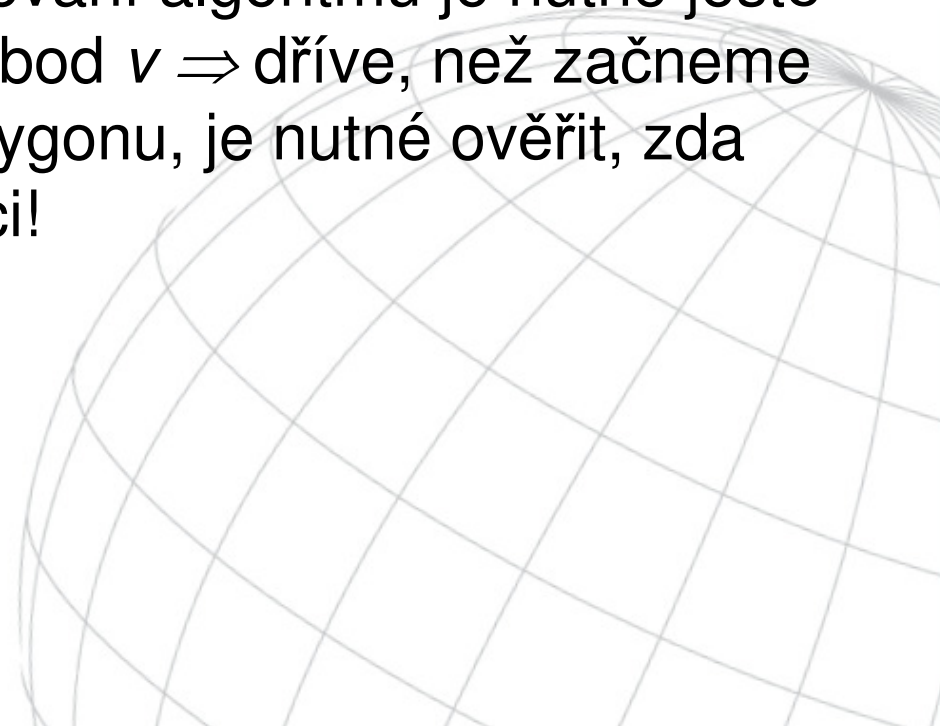
Lokace bodu

- V předchozím případě jsme neuvažovali některé singularity
→ uvažujme např. bod p_3 .
- Jednoduchým počítáním průsečíků bychom dostali číslo 9,
což nás vede k chybnému závěru!



Lokace bodu

- Řešení předchozí singularity je následující:
 - 1) neuvažujeme hrany kolineární s l ,
 - 2) průsečík hrany e s l je započítán pouze v případě, že hrana e má právě jeden koncový bod nad l .
- Pro úplnou správnost fungování algoritmu je nutné ještě uvažovat jeden případ, viz bod $v \Rightarrow$ dříve, než začneme testovat, zda bod leží v polygonu, je nutné ověřit, zda neleží přímo na jeho hranici!

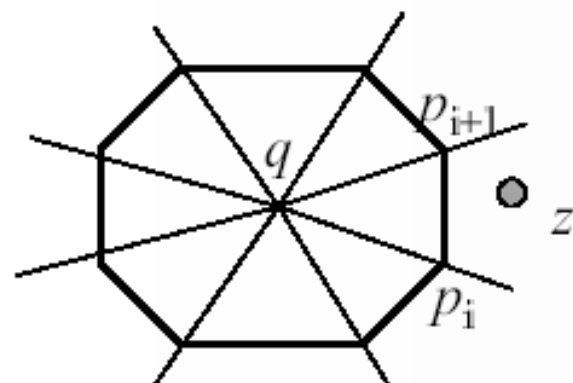


Lokace bodu

- **Půlení intervalu** (pro konvexní polygony).
- Mějme polygon P , vnitřní bod q (ten je nejprve nutné v rámci předzpracování najít).
- Vrcholy polygonu uspořádané.

■ Alg.:

- 1) binární vyhledávání podle úhlu.
- 2) nakonec porovnání vůči jedné hraně.



Lokace bodu

- Zpracování:
 - bod z leží mezi vrcholy p_i a p_{i+1} $\Leftrightarrow$ úhel zqp_{i+1} je pravotočivý, úhel mezi zqp_i je levotočivý.

Test pravotočivosti:

$$p_i = (x_i, y_i)$$

$$\det \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}$$

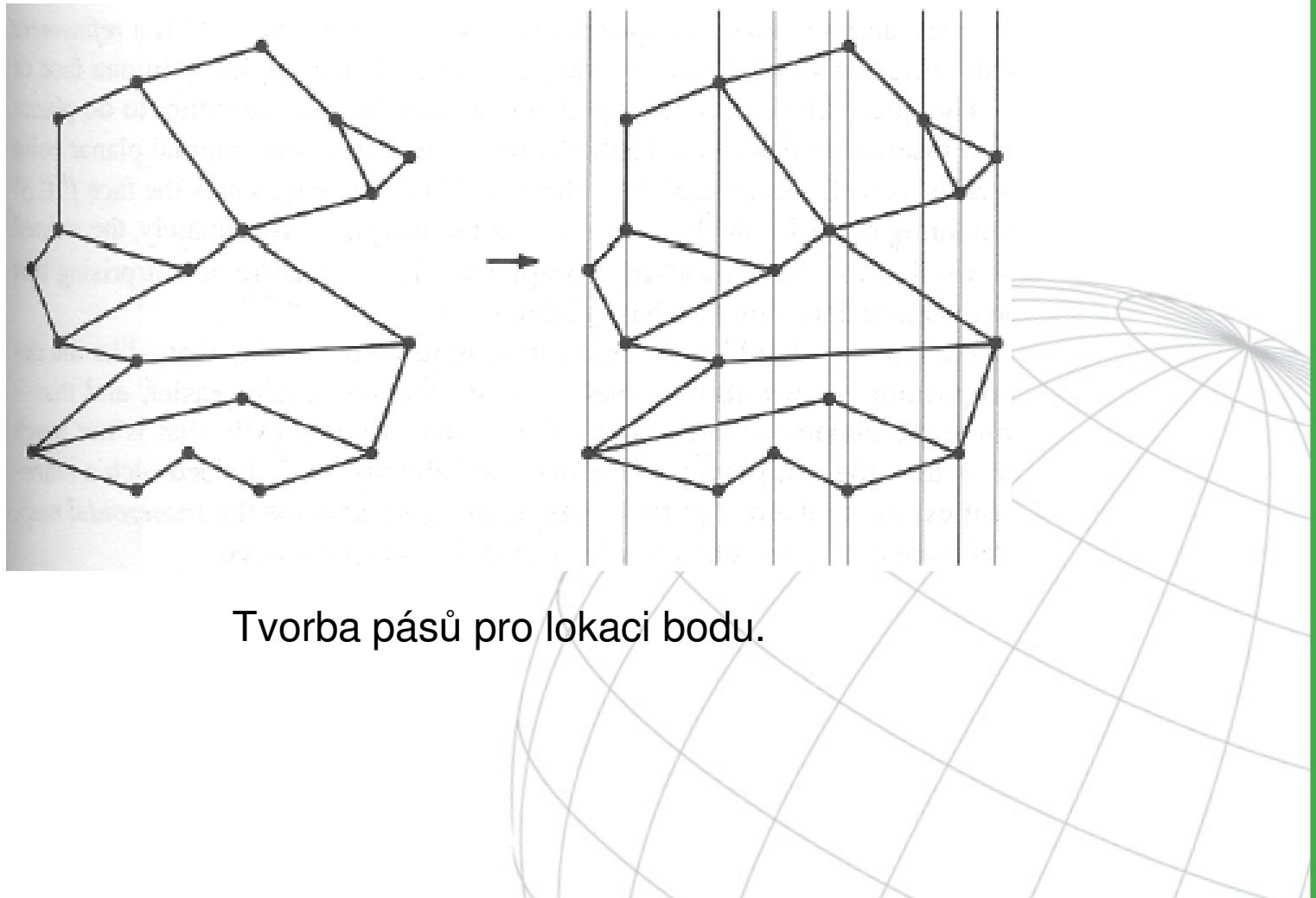
$p_1 p_2 p_3$ je pravotoč. $\Leftrightarrow \det. < 0$

Lokace bodu

- **Metoda pásů.**
- VSTUP: množina polygonů, bod p .
- ÚKOL: určit, ve kterém z polygonů se bod p nachází.
- POSTUP:
 - Vytvoření svislých, nebo vodorovných pásů.
 - X souřadnice seřadit.
 - V pásu úsečky seřadit podle y .
- Dotaz: $O(\log n)$.



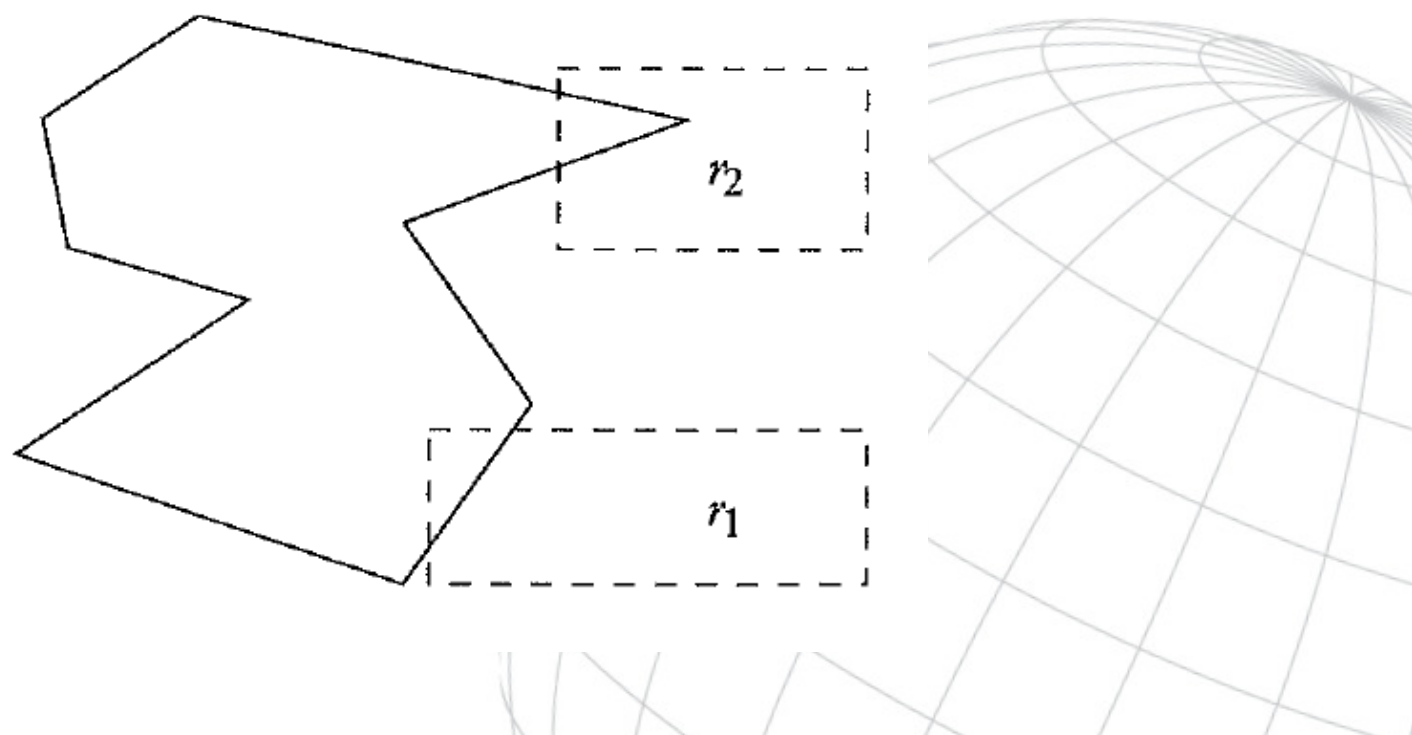
Lokace bodu



Tvorba pásů pro lokaci bodu.

„windowing“

- „windowing“ → výběr oknem
- V podstatě jde o testování, zda má geometrický objekt průnik s dotazovacím oknem (obdélníkem).



„windowing“

- Testovat, zda nějaký vrchol polygonu leží uvnitř dotazovacího okna, není efektivní – polygon může mít s oknem průnik, aniž by okno obsahovalo nějaký vrchol polygonu.



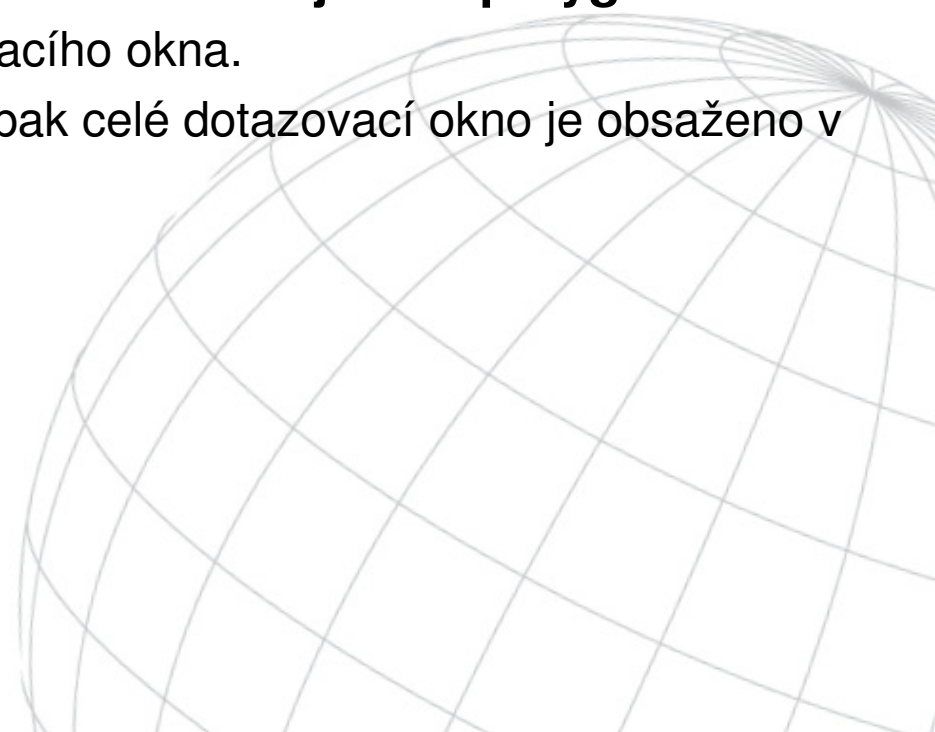
„windowing“

- Z tohoto důvodu se prochází jednotlivé hrany polygonu a testuje se, zda má hrana s oknem neprázdný průnik.
- Musíme také ale ohlídat případy, kdy je dotazovací okno zcela obsaženo nějakým objektem, nebo dotazovací okno obsahuje celý objekt → pokud nenajdeme žádný průnik mezi hranou polygonu a oknem, musíme provést dva testy na vztah „obsahuje“.



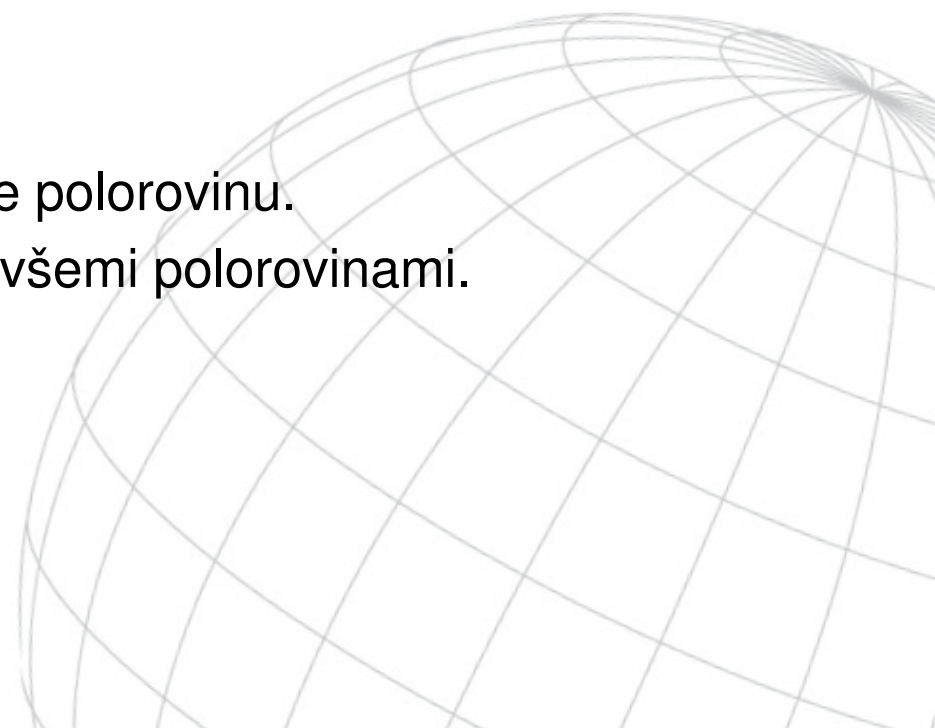
„windowing“

- Testování na vztah „obsahuje“:
 - **Je polygon celý obsažen v dotazovacím okně?**
 - p = libovolný bod polygonu.
 - pokud p je v dotazovacím okně $\Rightarrow$ polygon je uvnitř okna.
 - **Je dotazovací okno celé obsaženo v nějakém polygonu?**
 - p = libovolný vrchol dotazovacího okna.
 - pokud p je uvnitř polygonu, pak celé dotazovací okno je obsaženo v příslušném polygonu.
- $O(n)$.



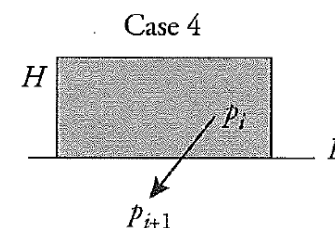
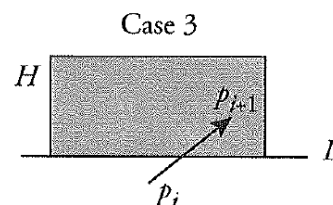
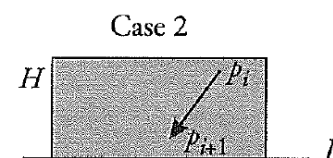
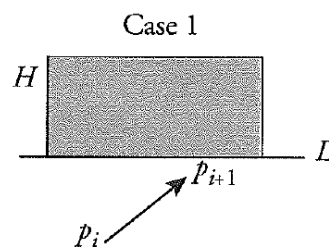
„clipping“

- V případě „clipping“ budeme chtít vrátit geometrii oříznutou výběrovým oknem.
- Vstup: okno W , polygon P .
- Výstup: polygon P' vzniklý oříznutím polygonu P oknem W .
- Postup:
 - Každá hrana okna W definuje polorovinu.
 - Polygon postupně ořízneme všemi polorovinami.



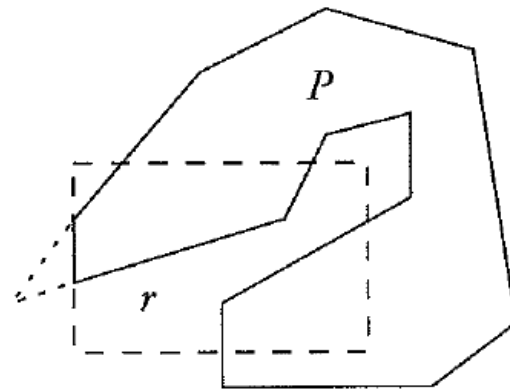
„clipping“

- Uvažujme následující obrázek:
- Hrana e_i je mimo $H \rightarrow$ bod p_{i+1} je odstraněn ze seznamu vrcholů P .
- Hrana e_i je uvnitř $H \rightarrow p_{i+1}$ zůstane zachován.
- Hrana e_i protíná L a p_{i+1} je uvnitř $H \rightarrow p_{i+1}$ zůstane zachován a průsečík L a e_i je přidán do seznamu vrcholů polygonu P .
- Hrana e_i protíná L a p_{i+1} je mimo $H \rightarrow p_{i+1}$ je odstraněn a průsečík L a e_i je přidán do seznamu vrcholů polygonu P .

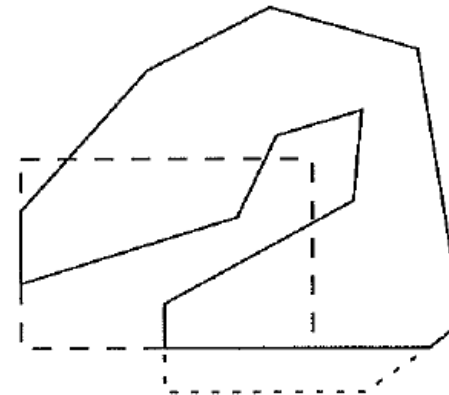


„clipping“

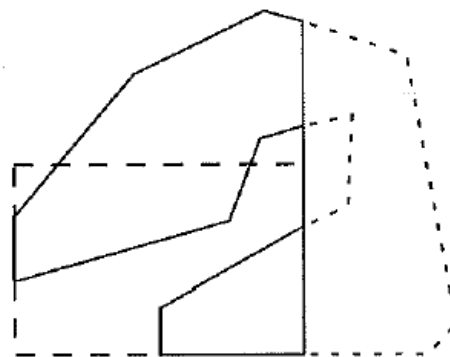
- $O(n)$.



(a)



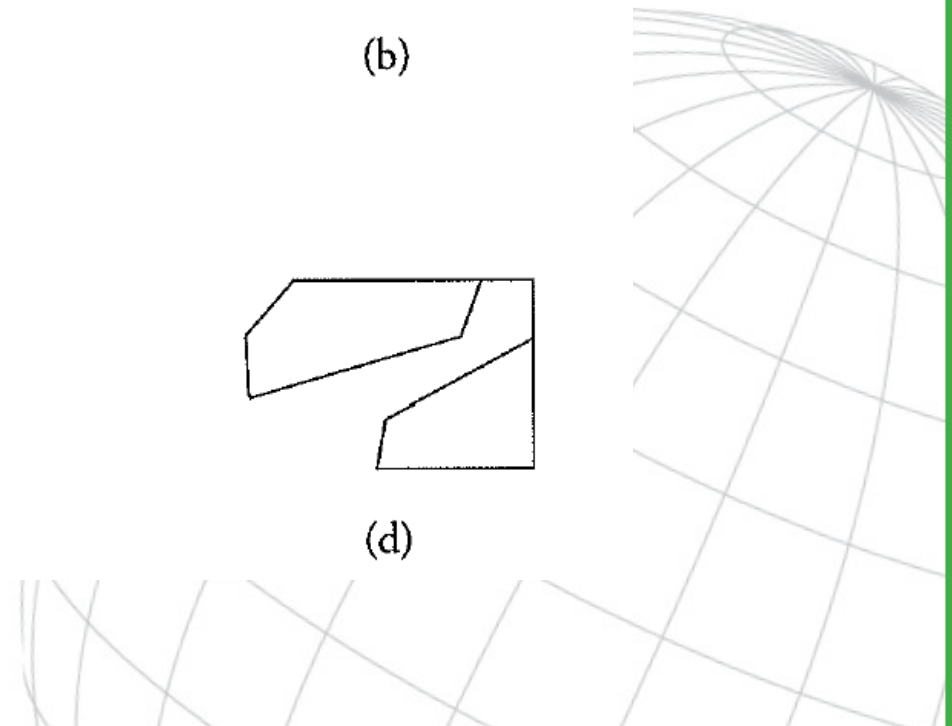
(b)



(c)

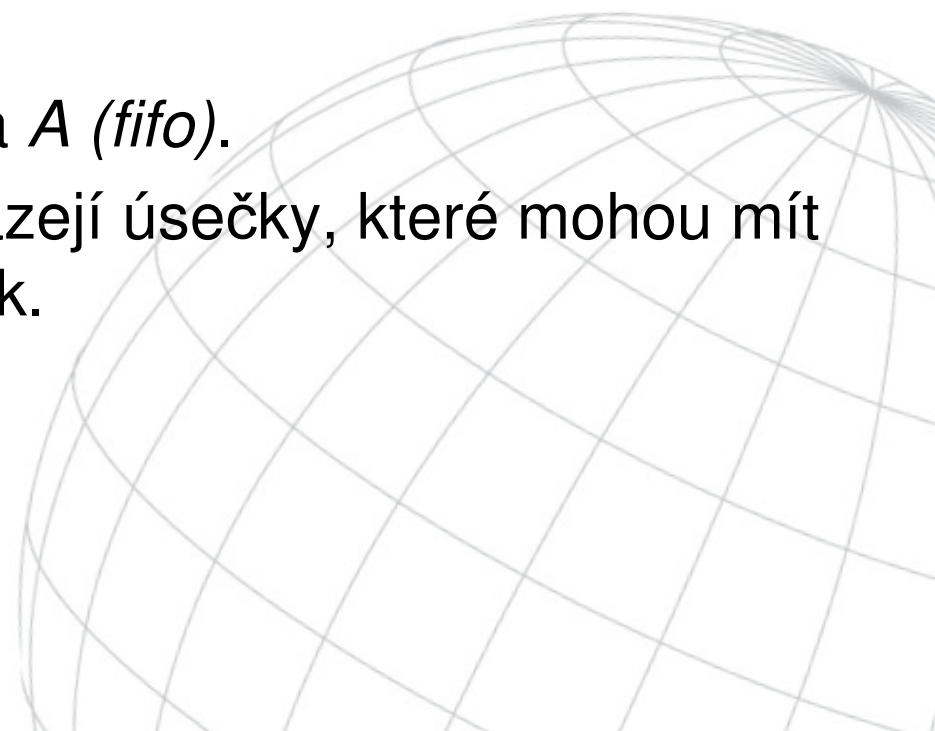


(d)



Nalezení průsečíků linií

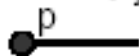
- **Sweep – line algoritmus.**
- Vstup: Množina úseček S .
- Výstup: Množina všech průsečíků.
- Vrcholy úseček je nutné seřadit podle x a vložit do tzv. časového plánu δ .
- Pomocná struktura – fronta A (*fifo*).
- Struktura T – v ní se nacházejí úsečky, které mohou mít s aktuální úsečkou průsečík.



Sweep – line algoritmus

- $A \leftarrow 0$; **while** $\delta \neq 0$ **do** { do vyčerpání čas. plánu δ }

- vyber z δ jeden vrchol p

- **if**  **then**

- vlož s do T

- najdi v T s_1 bezprostředně nad s

- najdi v T s_2 bezprostředně pod s

- **if** $s_1 \cap s$ **then** $A \leftarrow A + (s_1, s)$; **if** $s_2 \cap s$ **then** $A \leftarrow A + (s, s_2)$

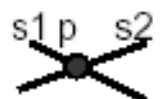
- **else if**  **then**

- najdi v T s_1 bezprostředně nad s

- najdi v T s_2 bezprostředně pod s

- **if** $s_1 \cap s_2$ **then** $A \leftarrow A + (s_1, s_2)$

- zruš s v T

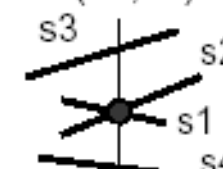
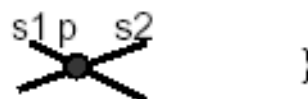
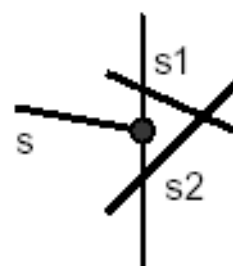
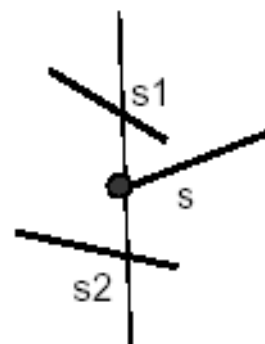
- **else** {  }

- $S_3 \leftarrow$ úsečka bezprostředně nad s_1 , s_4 pod s_2

- **if** $s_3 \cap s_2$ **then** $A \leftarrow A + (s_3, s_2)$; **if** $s_1 \cap s_4$ **then** $A \leftarrow A + (s_1, s_4)$

- vyměň s_1 , s_2 v T

- **endif**



Sweep – line algoritmus

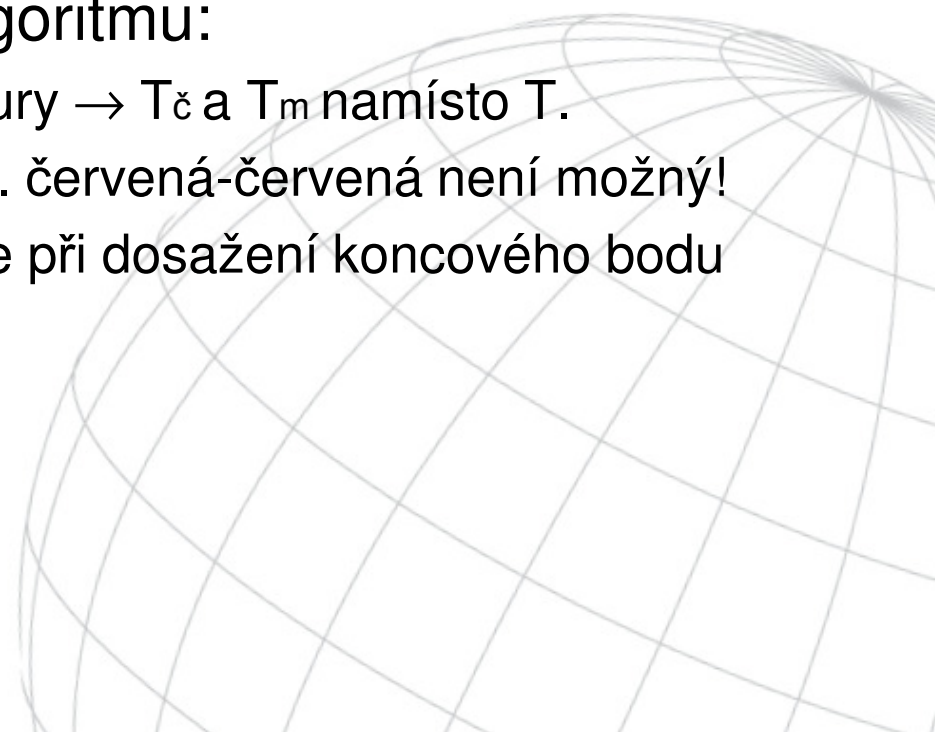
- { zpracování detekovaných průsečíků }
 - **while** $A \neq 0$ **do**
 - $(s, s') \leftarrow A$ { mají souřadnici x }
 - **if** x není prvkem δ **then**
 - označ (s, s')
 - vlož x do δ
 - **endif**
 - **enddo**
- **enddo**

- $O((n+k)\log n)$



„red – blue“ problém

- Vstup: Dvě množiny úseček, jedny červené (red), druhé modré (blue).
- Výstup: Všechny průsečíky mezi červenými a modrými úsečkami.
- Modifikace předchozího algoritmu:
 - Budeme udržovat dvě struktury $\rightarrow T_{\text{č}}$ a T_{m} namísto T .
 - Průsečík modrá-modrá, resp. červená-červená není možný!
 - Ke změně $T_{\text{č}}$, resp. T_{m} dojde při dosažení koncového bodu úsečky příslušné barvy.
- $O(n \log n + k)$.



„red – blue“ problém

- Polylinie – složena z více neprotínajících se úseček.
- Pokud bychom řešili průsečík(y) dvou polylinií, lze použít algoritmus na řešení „red – blue“ problému.



Konzistence prostorových dat

- Obecné požadavky na tzv. topologicky čistá data:
 - Každá hrana musí být ohraničena dvěma uzly (počáteční a koncový uzel).
 - Každá hrana má stěnu vlevo a stěnu vpravo.
 - Každá stěna má uzavřenou hranici sestávající ze střídavé posloupnosti uzlů a hran.
 - Kolem každého uzlu je střídavá uzavřená sekvence hran a stěn.
 - Hrany se vzájemně neprotínají, pouze v uzlech.

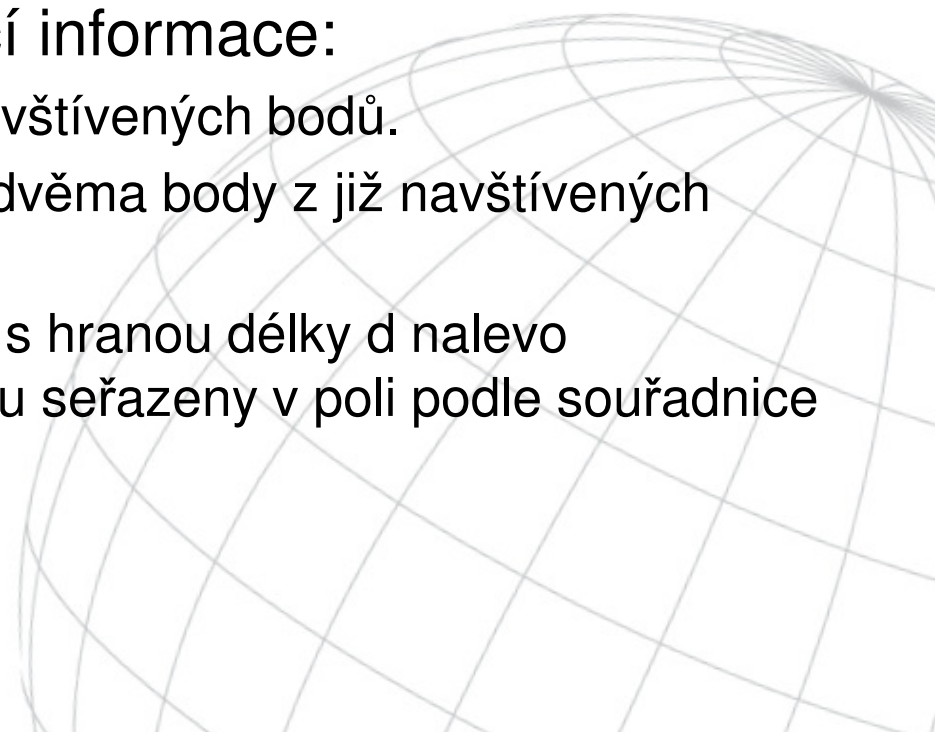
Konzistence prostorových dat

- Orientace na data (DKM) v ISKN:
 - Duplicitní mapové značky (1).
 - Křížení linií vnitřní kresby – nespočítané průsečíky (2).
 -
- Využití výpočetní geometrie:
 - (1) → Nalezení dvojice/dvojic nejbližších bodů.
 - (2) → Nalezení všech průsečíků v dané množině hran.

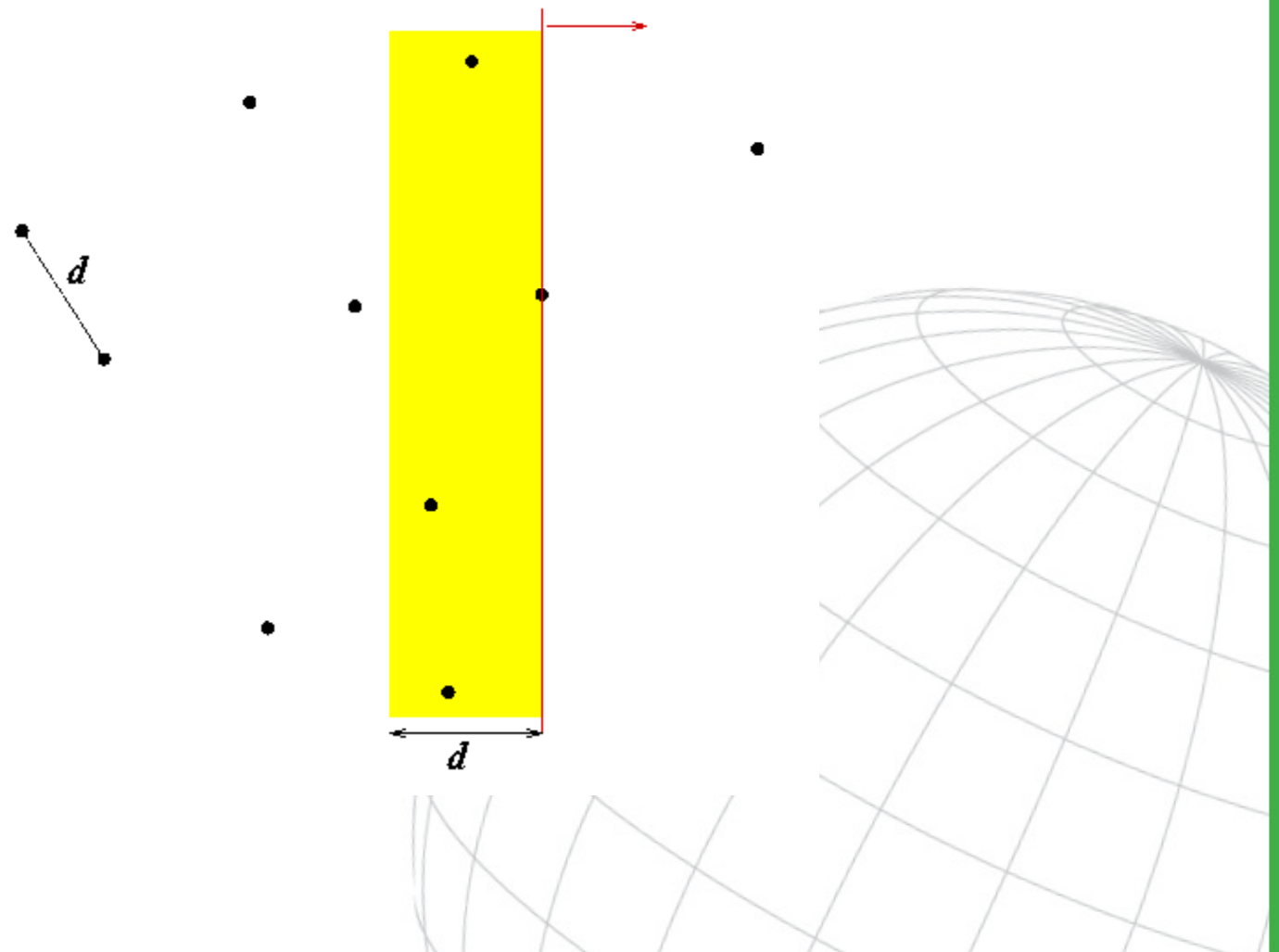


Dvojice nejbližších bodů

- Vstup: Množina n bodů v rovině.
- Výstup: Dvojice nejbližších bodů.
- Technika: použití sweep-line.
- Při pohybu sweep-line skrz množinu bodů v každém bodě udržujeme následující informace:
 - Nejbližší dvojici bodů z již navštívených bodů.
 - Nejkratší vzdálenost d mezi dvěma body z již navštívených bodů.
 - Všechny body uvnitř řetězce s hranou délky d nalevo od sweep-line. Tyto body jsou seřazeny v poli podle souřadnice y .

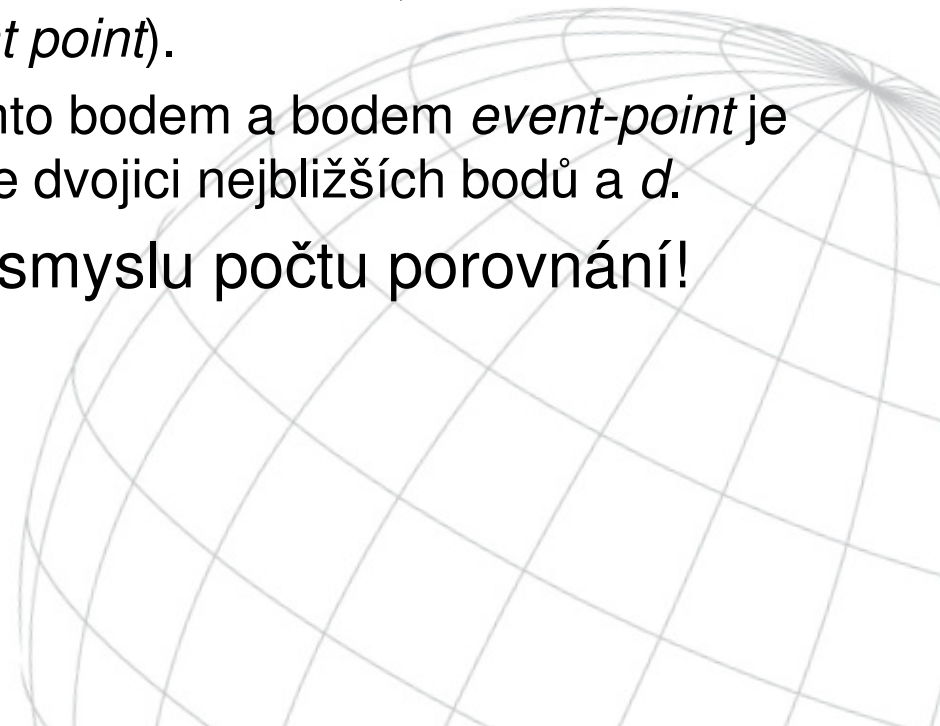


Dvojice nejbližších bodů

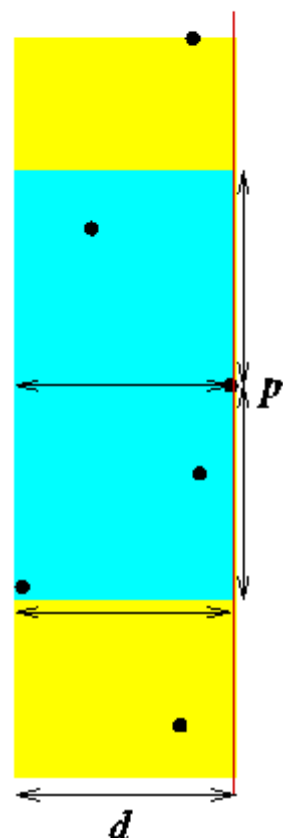


Dvojice nejbližších bodů

- V každém bodě, ve kterém se zastaví sweep-line, provedeme následující operace:
 - 1) Odstraníme ze seřazeného pole bodů všechny body, které jsou ve vzdálenosti větší než d .
 - 2) Ze zbylých bodů určíme nejbližší bod k bodu, ve kterém se zastavila sweep-line (*event point*).
 - 3) Pokud vzdálenost mezi tímto bodem a bodem *event-point* je menší než d , aktualizujeme dvojici nejbližších bodů a d .
- Bod 2) lze vylepšit → ve smyslu počtu porovnání!



Dvojice nejbližších bodů



- Uvažujeme pouze body, které leží ve vzdálenosti menší d v obou osách!
- $O(n \log n)$.



Zdroje

- JANEČKA, K.: [Zajištění konzistence prostorových dat v Informačním systému katastru nemovitostí](#). In: Proceedings of GIS Ostrava 2008. Tanger. Ostrava, 2008. s. 1-8. ISBN 978-80-254-1340-1.
- KOLINGEROVÁ, I.: [Přednášky k předmětu Vybrané algoritmické metody](#). FAV ZČU v Plzni.
- MURRAY, Ch.: [Oracle Spatial Developer's Guide, 11g Release 1 \(11.1\)](#). Oracle. 2009.
- POKORNÝ, J.: [Prostorové datové struktury a jejich použití pro indexaci prostorových objektů](#). In: Proceeding of GIS Ostrava 2000. Ostrava, 2000.
- POKORNÝ, J.: [Prostorové objekty a SQL](#). In. Proceedings of GIS Ostrava 2001. Ostrava, 2001. ISSN: 1213-239X.
- POKORNÝ, J.; ŽEMLIČKA, M.: [Základy implementace souborů a databází](#). Karolinum. Praha, 2004. 211 s. ISBN: 80-246-0837-5.
- ŽEMLIČKA, M.: [Přednášky k předmětu Organizace a zpracování dat II](#). MFF UK v Praze.

