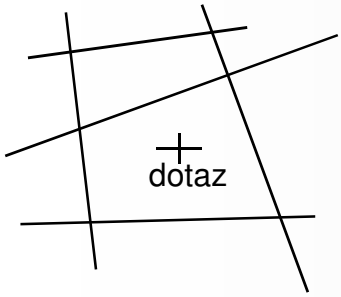
Geometrické vyhledávání

Ivana Kolingerová

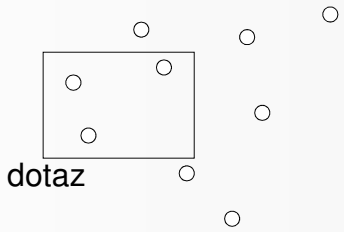
Obsah:

1. Lokace bodu
   1. Konvexní polygony
   2. Obecné polygony
   3. Planární dělení
2. Hledání intervalů (range search)

Lokace



Hledání intervalu



Někdy vzájemný převod

Aplikace: nejbližší sousedé,
lokace myši …

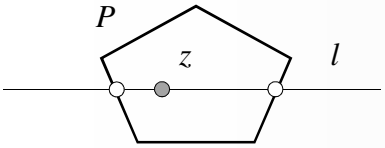
Aplikace: databáze (multiklíč. vyhl.
zeměpis, statistika,
návrh automatů

AVG2-3

Lokace bodu

1. Konvexní polygon
   - bez předzpracování: *O(N)*

a) Ray crossing



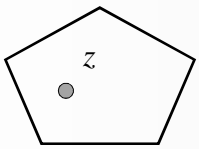
if počet průsečíků *l* s *P*
doleva od *z* je lichý **then**
uvnitř
else
vně

Pozor na singulární případy!

AVG2-4

b) Test polohy bodu vůči každé hraně

- Obecná rovnice přímky, dosadit bod, testovat znaménko výsledku
- Pozor na orientaci (obrácená otočí znaménka)
- Ale: zbytečně mnoho výpočtů pro nalezení rovnic – potřebujeme jen znaménka – raději orientační test



AVG2-5

Orientační test (predikát):

$p_i = (x_i, y_i)$

$$\det\begin{bmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{bmatrix}$$

$p_1\ p_2\ p_3$ je pravotoč. <=> det < 0

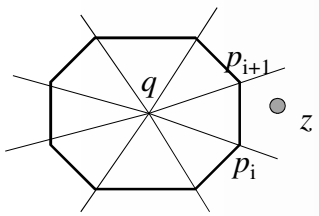
(a naopak)

Nuly: singularity (body na přímce)

AVG2-6

- S předzpracováním: *O(N)* předzprac.,
O(log N) čas, *O(N)* paměť

c) Půlení intervalu



q – vnitřní bod
- binární vyhledávání podle úhlu
- pak porovnání vůči jedné hraně
- místo q může být vrchol

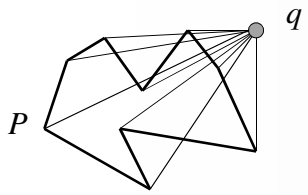
Předzpracování: nalezení *q*
uspořádání vrcholů do dat. struktur
pro půlení intervalu
(jsou v pořadí=>*O(N)*)

Zpracování: *z* leží mezi p_i a p_{i+1} <=>
úhel zqp_{i+1} je pravotoč., úhel zqp_i levotočivý

AVG2-7

2. Nekonvexní polygon
   - bez předzpracování: ray crossing

d) Ovíjení (winding number) *O(N)*



if *q e P* **then** 2π
else 0

asi 20x pomalejší než ray-crossing

Po úpravě i pro nekonvexní polyedr,
suma pak 0 nebo 4 π

AVG2-8

- S předzpracováním:
  - pro hvězdic. tvar viz e)
  - pro obecný tvar: rozdělit na konvexní části nebo hierarchie subpolygonů nebo e)

AVG2-9

e) Planární dělení

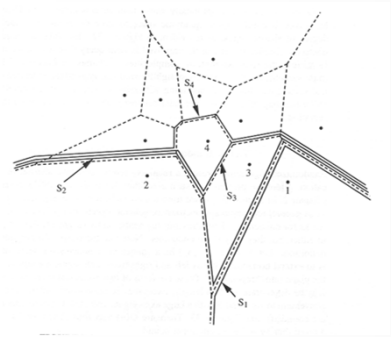
- s užitím speciálních vyhledávacích struktur je nejlepší současný výsledek *O(N)* předzpracování a paměť, *O(log N)* zpracování, ale alg. je ošklivý ☹ [Ede86]
- jednodušší: monotónní rovinné dělení (všechny hrany jsou monotónní ve stejném směru) [Ede86]

Příklad (Voronoi diagram)

Si – separátor (i=1,2,..) – monotónní, dělí rovinu na 2 části
=> možnost binárního vyhledávání

AVG2-10

Postup:
- Najít separátory (nepříjemné ☺)
- Binární vyhledávání vertikálně (mezi separátory)
- Binární vyhledávání horizontálně (v rámci separátoru)



O(N²) paměť (!)
Y *O(log N)*
X *O(log N)*

O(log² N) čas
O(N²) paměť

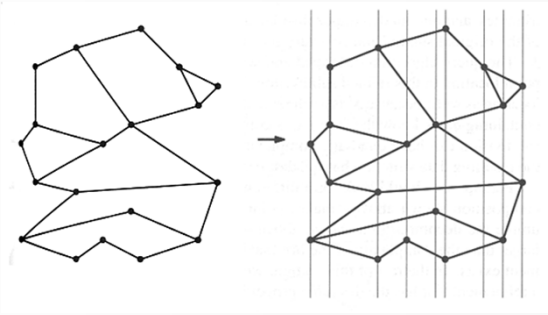
Není optimální, ale je jednodušší
Jde upravit na asymptoticky optimální (layered data structure, opakované využití výsledků půlení intervalu) [Ede86]

AVG2-11

Metoda pásů

- svislé nebo vodorovné pásy (slabs)
- x souřadnice seřadit v poli =>
  pás s hledaným bodem v *O(log N)*
- v pásu úsečky seřazeny podle y

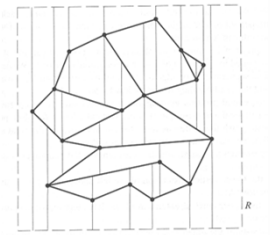
Viz např. [ORo94]



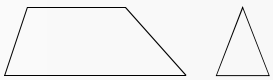
Konstrukce *O(N log N)*
Dotaz *O(log N)*
Paměť *O(N^2)* ! (v praxi méně, ale ...)

AVG2-12

Lichoběžníková mapa (trapezoidal map)



viz např. [ORo94]



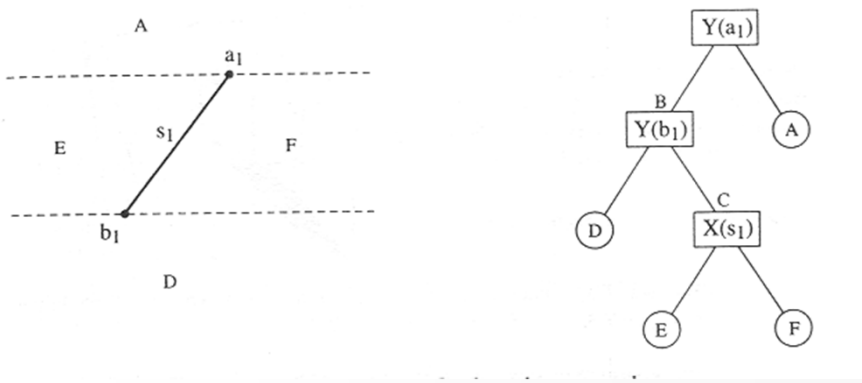
- každý lichoběžník sousedí
  max. s 2 lichoběžníky "nad" a 2 "pod" –
  sousedi sdílejí část vodorovné hrany =>
  možné binární vyhledávání
- vyhledávací strom
- uzly: listy s lichoběžníky, vnitřní s x nebo s y

Konstrukce inkrementální, randomizovaná

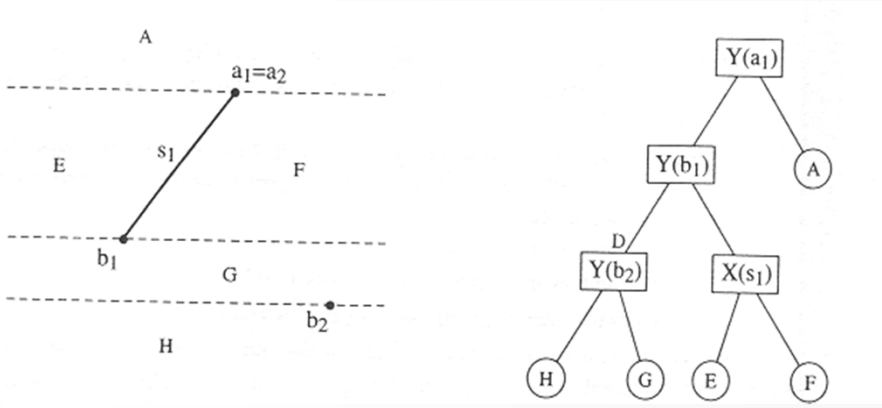
Konstrukce	*O(N log N)* očekávané
Očekávaná výška	*O(log N) (ale Ω(N)) !!*
Paměť	*O(N)*

AVG2-13

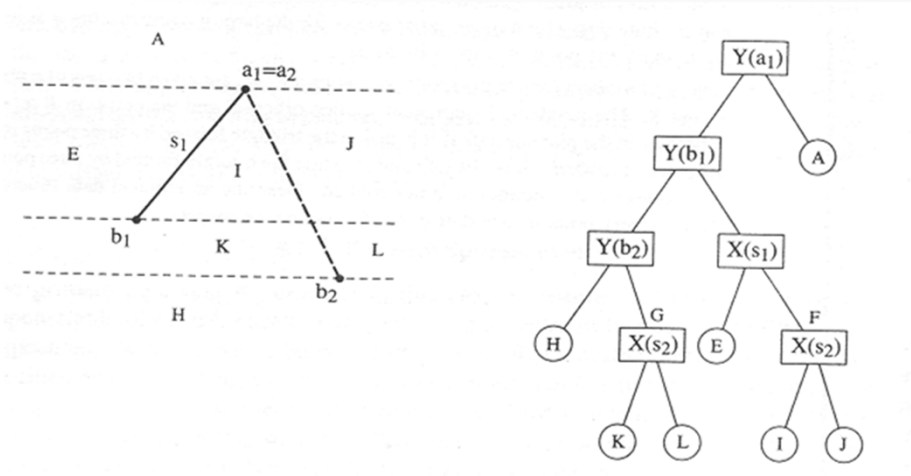
Lichoběžníková mapa (trapezoidal map)



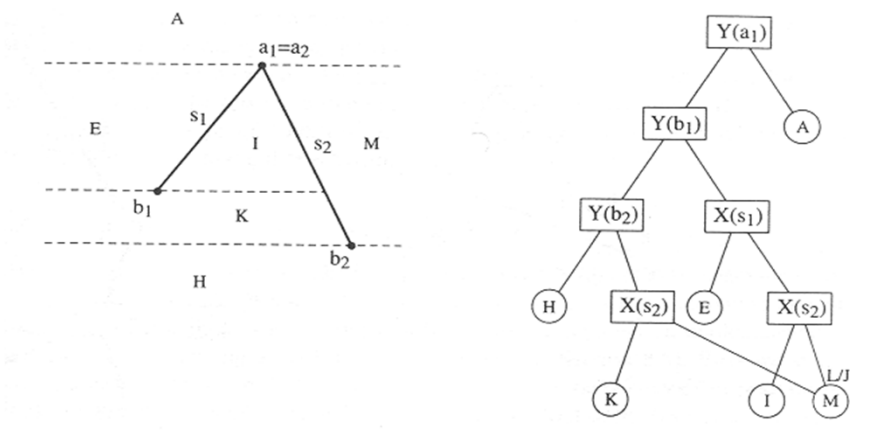
AVG2-14



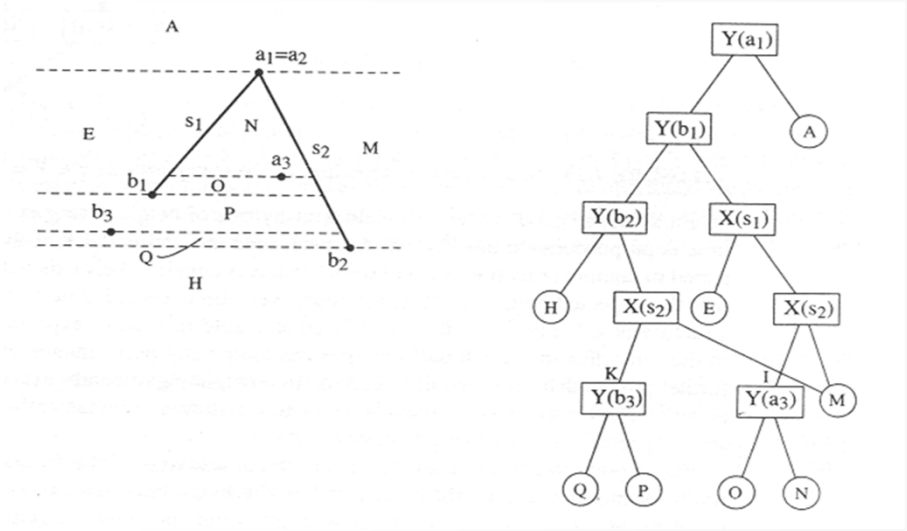
AVG2-15



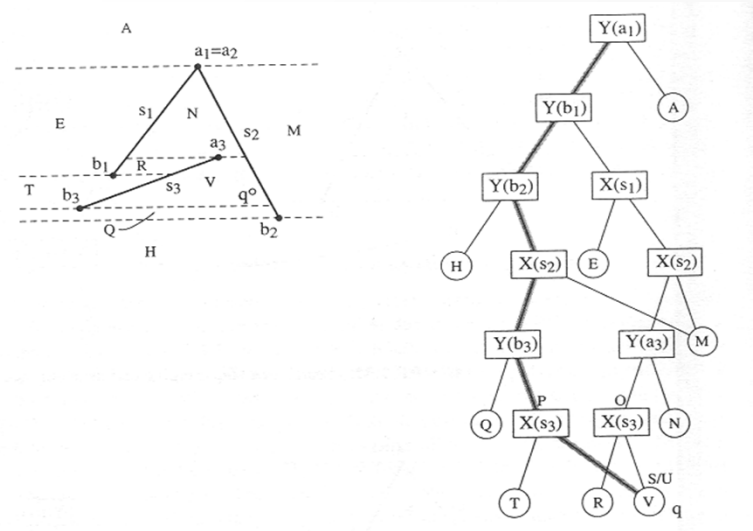
AVG2-16



AVG2-17



AVG2-18



Odkazy:

[Ede86] Edelsbrunner H., Guibas L.J.:Optimal Point Location in a Monoton Subdivision, *SIAM J. Comput.*, 15(2), 317–340, http://epubs.siam.org/doi/abs/10.1137/0215023, http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-2.pdf

[ORo94] O'Rourke J.: Computational Geometry in C, Cambridge University Press, 1994 (first edition), 1998 (second edition)

Also:
de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry. Algorithms and Applications, Springer-Verlag (1997)

https://www.cs.umd.edu/~mount/Papers/crc05-prox.pdf