# ORTOGONÁLNÍ VYHLEDÁVÁNÍ (1)

P množina bodů v rovině

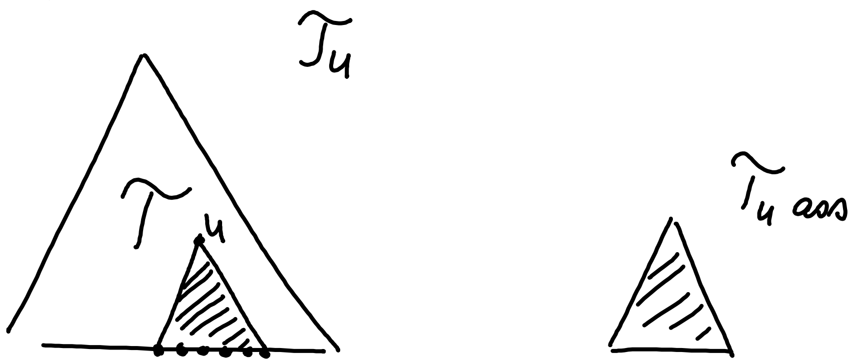
najít vhodnou strukturu, která umožní pro zadaný pravoúhelník

$$[x, x'] \times [y, y']$$

vyhledání bodů z P, které v něm leží

kd-stromy

range trees




T je uspořádání podle x-ové souřadnice

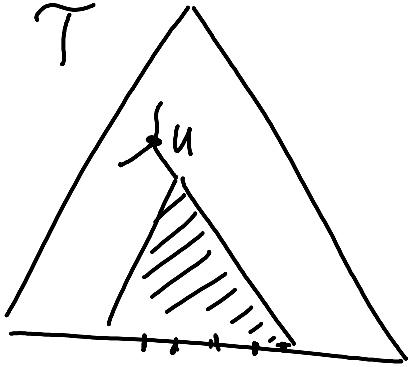
uspořádání podle y

2

| | range trees | kd-stromy |
|---|---|---|
| Paměťová náročnost | $O(n \log n)$ | $O(n)$ |
| Čas konstrukce | $O(n \log n)$ | $O(n \log n)$ |
| Vyhledávání | $O(\log^2 n + k)$ | $O(n^{1/2} + k)$ |

k počet nalezených

T




prohledávání podstromu

$O(\log n + k_u)$

projdeme $\sim O(\log n)$ uzlů

$$\sum_{u \text{ na cestě}} (\log n + k_u) \cong \log n \cdot \log n + (k_{u_1} + k_{u_2} + \dots) = \log^2 n + k$$

3

## Obecně v dimenzi d

| | range trees | kd-stromy |
| --- | --- | --- |
| paměť | $O(n \log^{d-1} n)$ | $O(n)$ |
| konstrukce | $O(n \log^{d-1} n)$ | $O(n \log n)$ |
| čas vyhledávání | $O(\log^{d} n + k)$ | $O(n^{1-\frac{1}{d}} + k)$ |

4

Vše jsme dělali za předpokladu, že pro každé dva body $p, q \in$

$p_x \neq q_x$ a $p_y \neq q_y$

Jak toto odstraníme? Následujícím trikem.

Místo $p_x$ bereme dvojici $(p_x, p_y)$ v lexikograf. uspořádání
prvně podle x, pak podle y.

Místo $p_y$ bereme dvojici $(p_y, p_x)$ v lexikograf. uspořád
prvně podle 1. složky, pak podle

5

Súčin intervalů $\quad Q = [x, x'] \times [y, y']$

nahradíme množinou

$$Q' = \big[(x, -\infty), (x', \infty)\big] \times \big[(y, -\infty) \times (y', +\infty)\big]$$

Tato volba má tuto vlastnost

$$(p_x, p_y) \in Q \iff \big((p_x, p_y), (p_y, p_x)\big) \in Q'$$

$$\Leftarrow \quad \boxed{x \le p_x \le x'} \quad \boxed{y \le p_y \le y'}$$

$$-\infty \le p_y \le \infty \qquad -\infty \le p_x \le \infty$$

6

# LOKALIZACE BODU

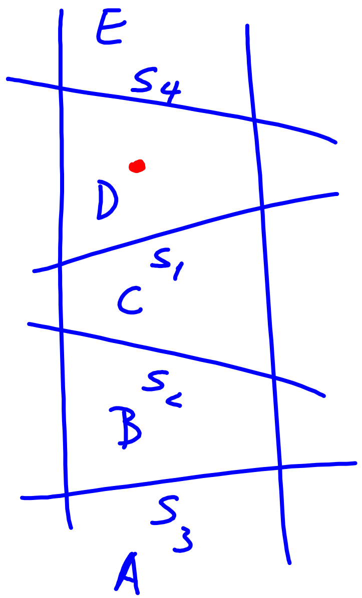
Máme mapu, dostaneme souřadnice bodu a naším úkolem je zjistit, ve kterém státě leží.

Máme rovinné podrozdělení. Chceme najít vyhledávací strukturu, která pro zadaný bod najde (stát), ve které tento bod leží.
oblast

(7)

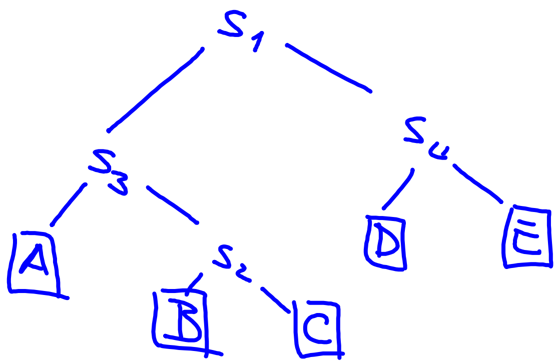
## Myšlenka

Pás roviny rozdělený na oblasti



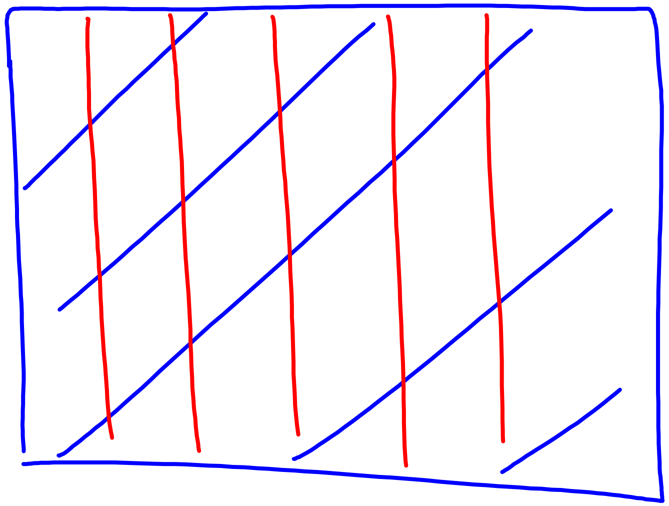

vyhledávání bodu je jednoduché

- lze použít „přirozené" uspořádání podle y-ové souřadnice

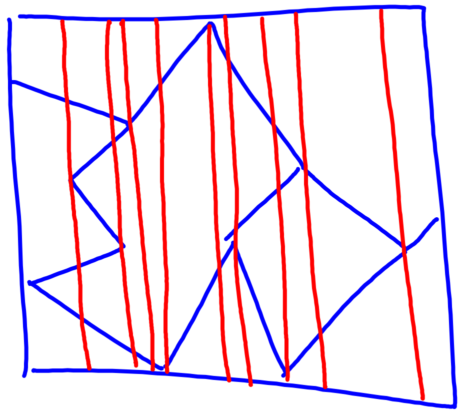


8

Chceme mapu dále rozdělit na podélné pásy



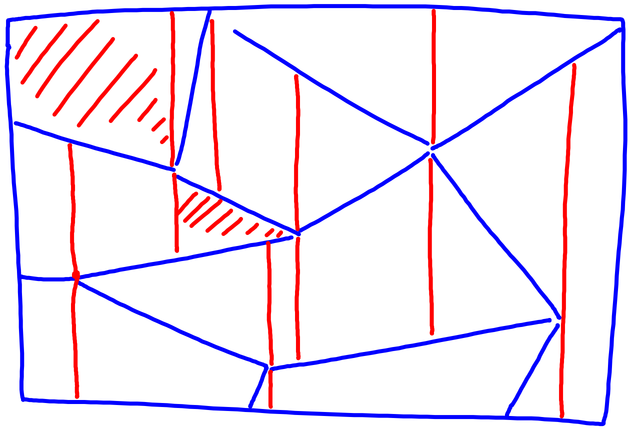
Tímto způsobem z podrozdělení o $n$ oblastech dostaneme podrozdělení obecně o $n^2$ oblas



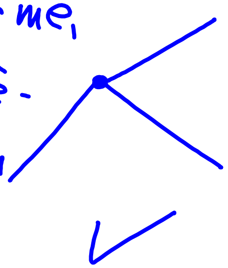
Vertikály vyplníme v konc. bo úseček.

9

# Lepší způsob

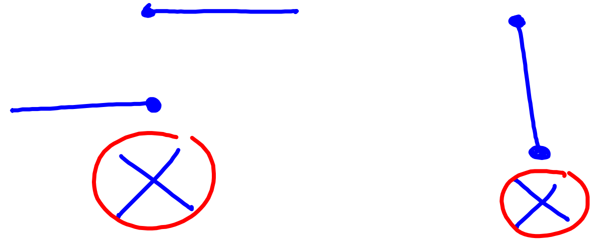


Navíc předpokládáme, že celé podrozdělení je v nějakém pravoúhelníku.
R

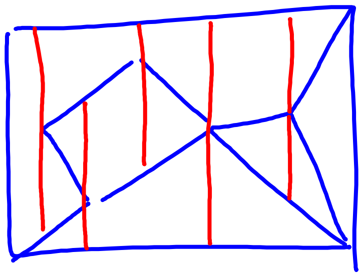


Každým koncovým bodem úsečky vedeme vertikální úsečku, k nejbližší úsečce nad a nejbližší úsečce pod naší úsečkou.

(Předpokládáme, že koncové body úseček mají různé x-ové souřadnice



10



R

Výsledkem této konstrukce je takzvaná lichoběžníková mapa

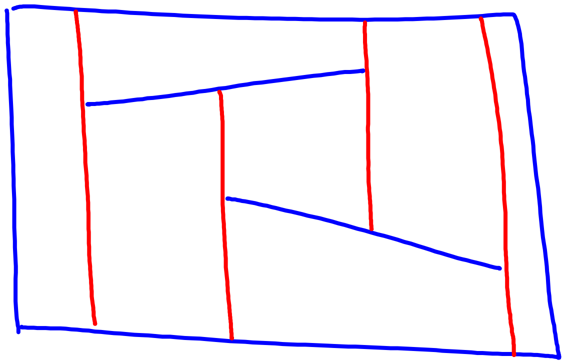
Počet lichoběžníků bude řádově stejný jako počet úseček

Pro konstrukci lichoběžníkové mapy a s ní související vyhledávací struktury jsou podstatné pouze hrany našeho rovinného podrozdělení. Tedy lichoběžníkovou mapu můžeme definovat a konstruovat pro množinu úseček v rovině.

11

Dána množina úseček v rovině $S=\{s_1, s_2, \dots, s_n\}$, $s_i \subseteq R$ pravoúhelník

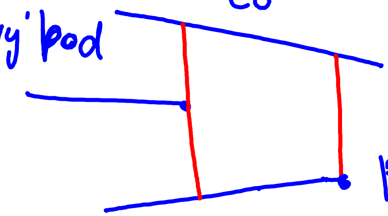
Lichoběžníková mapa je rozdělení $R$ na lichoběžníky tak, že každým konc. bodem úsečky vedeme vertikální úsečku k nejbližší vyšší a nižší úsečce



Čím je určen lichoběžník v mapě?

to

levý bod



pravý bod

Těmito 4 údaji je lichoběžník určen jednoznačně.

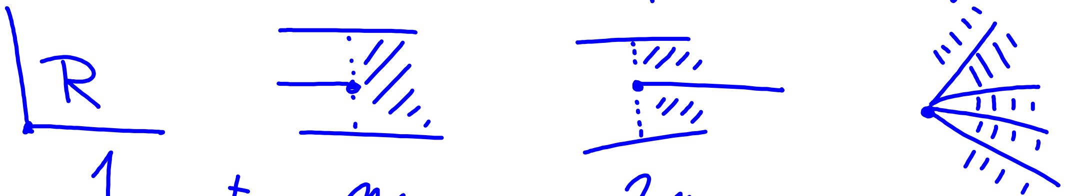
(12)

Lichoběžníková mapa pro $n$ úseček obsahuje nejvýše $6n+4$ vrcholů a $\underline{3n+1}$ lichoběžníků.

Dk: Počítání vrcholů:

$$\leq \underbrace{4}_{\text{vrcholy } R} + \underbrace{2n}_{\text{vrcholy úseček}} + \underbrace{2(2n)}_{\text{nové vrcholy}} = 6n+4$$

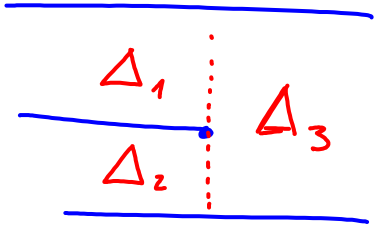
Počítání lichoběžníků podle toho, jak vypadá jejich levá strana




$$1 + n \qquad 2n \qquad = 3n+1$$

13

Sousední lichoběžníky ... mají společnou vertikální stranu




$\Delta_3$ je pravý horní soused $\Delta_1$
(mají společnou horní hranu)

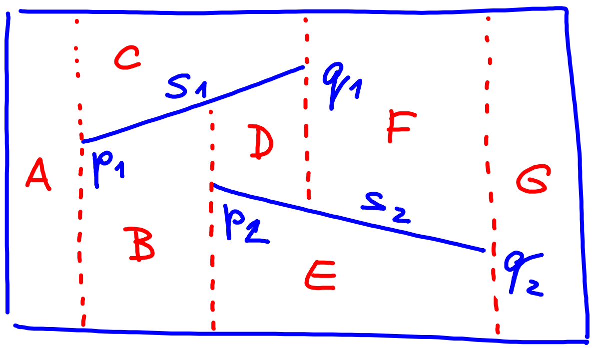
$\Delta_2$ je levý dolní soused $\Delta_3$
(mají společnou dolní hranu)

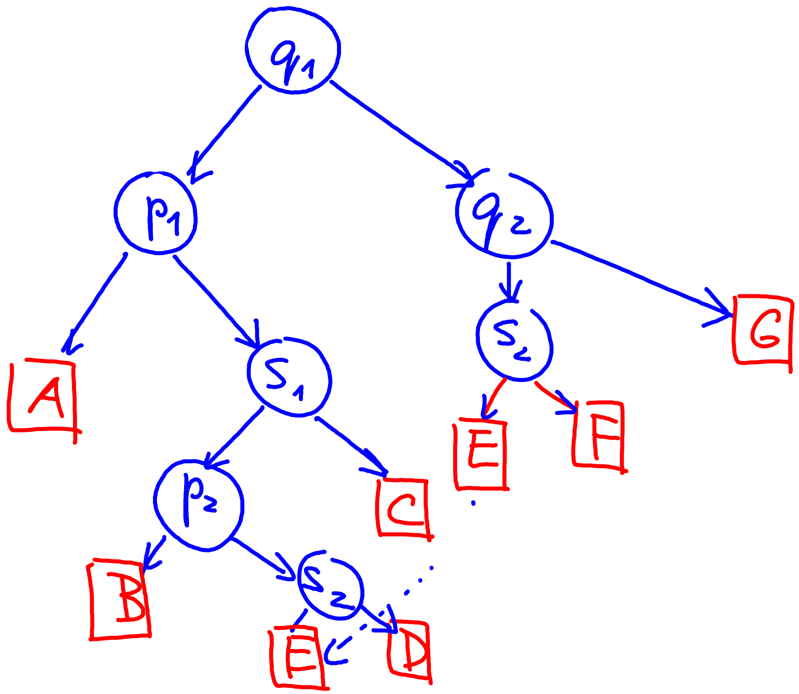
Lichoběžníková mapa pro množinu úseček $S$
$\mathcal{T}(S)$

V každém lichoběžníku uchováváme tyto informace
top, bottom, levý bod, pravý bod

14

Vyhledávací struktura pro lichoběžníkovou mapu
je orientovaný graf $\mathcal{D}$



15

Algoritmus pro vytváření lich. mapy a vyhledávací struktury je

- přírustkový
- může být prováděn náhodnostně

tj. mapu konstruujeme postupně pro množiny úseček

$$S_i = \{s_1, s_2, \dots s_i\}$$

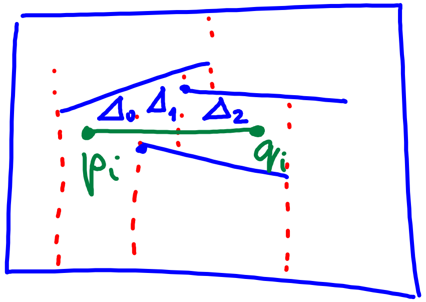
pro $i = 1, 2, \dots n$. Současně konstruujeme vyhledávací strukturu $\mathcal{D}(S_i)$.

Náhodnost spočívá v náhodném pořadí úseček

16

Rámový algoritmus - viz s. 28 v pseudokódech

$T(S_{i-1})$ přidáme úsečku $s_i$ do lichoběžníkové mapy




$s_i$ nová úsečka

$s_i$ prochází přes lichoběžníky $\Delta_0, \Delta_1, \Delta_2$

$\Delta_0$ najdeme pomocí vyhl. struktury jako lichoběžník, ve kterém leží bod $p_i$ (levý bod $s_i$), pokud tento již nebyl vrcholem jiné úsečky obsažen v mapě

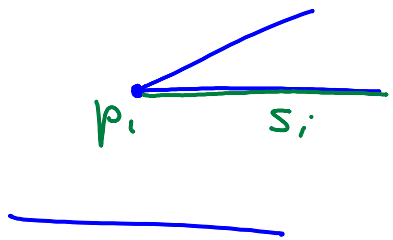
Jestliže pravý bod $q_i$ má x-ovou souřadnici větší než pravý bod $\Delta$ prochází $s_i$ dalším lichoběžníkem.

17

Za $l_1$ vezmeme pravého souseda $\Delta_0$. Kterého?

Vezmeme první bod pro $\Delta_0$. Je-li $s_i$ nad, vezmeme pravého horního souseda. Je-li $s_i$ pod, vezmeme pravého dolního souseda.

atd.




=> pseudokód podalgoritmu Follow Line Segment

str. 29