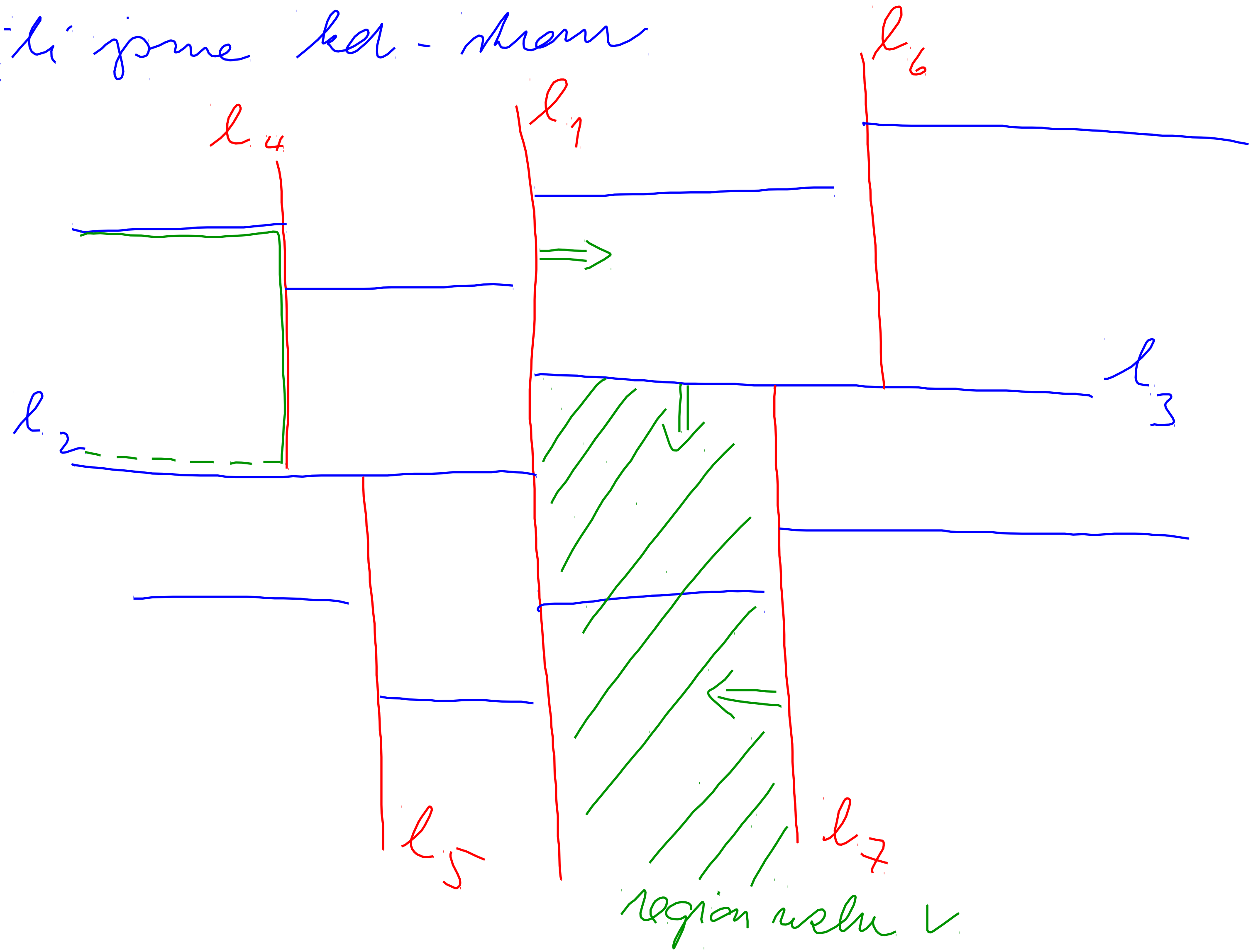
ORTOGONÁLNÍ VYHLEDÁVÁNÍ

V rovině máme množinu n bodů P

Sestrojili jsme kd-strom

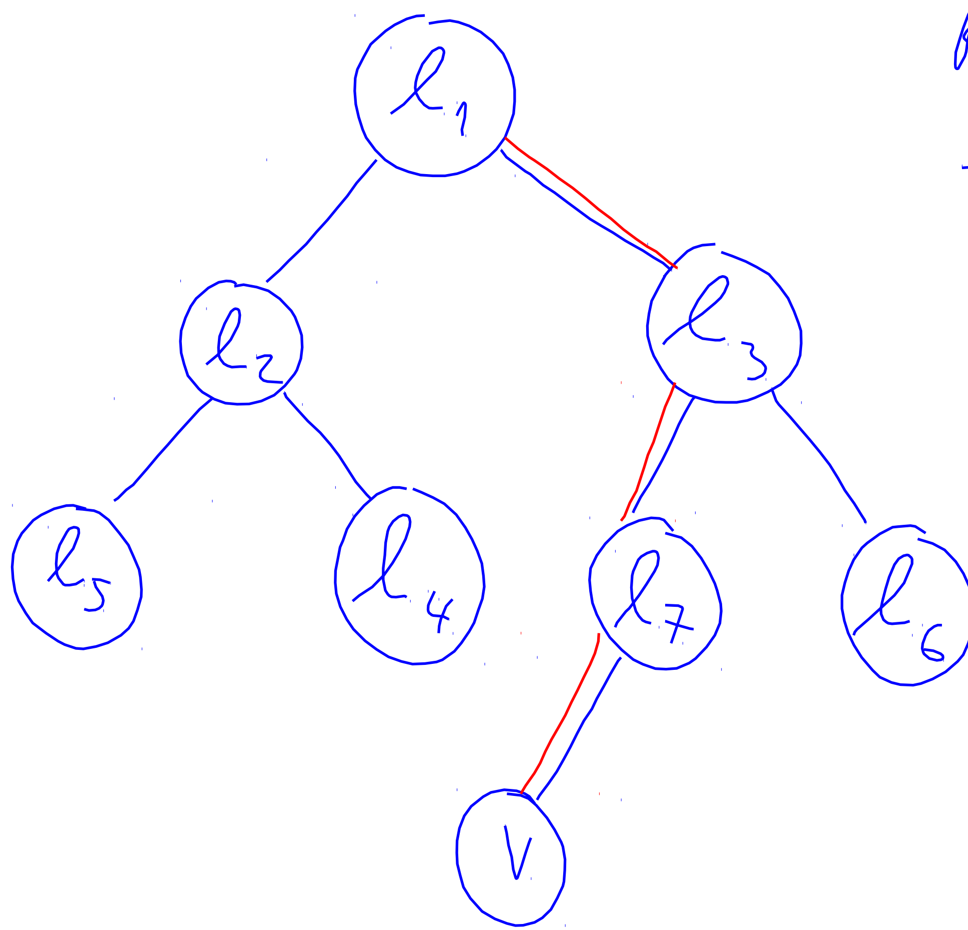


②

Vyhledávání, které body z P leží v pravoúhelníku

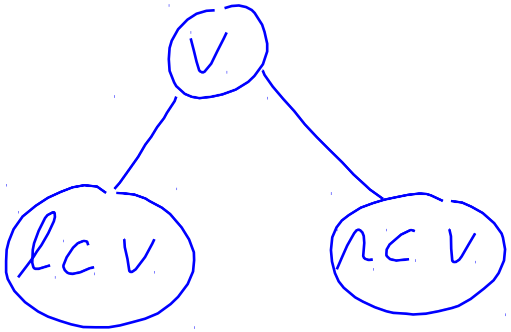
$$[x, x'] \times [y, y']$$

Potřebujeme pojem regionu uzlu



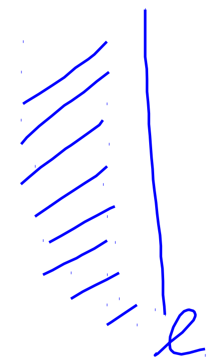
Region uzlu v = průnik polorovin určených cestou k uzlu v.

3



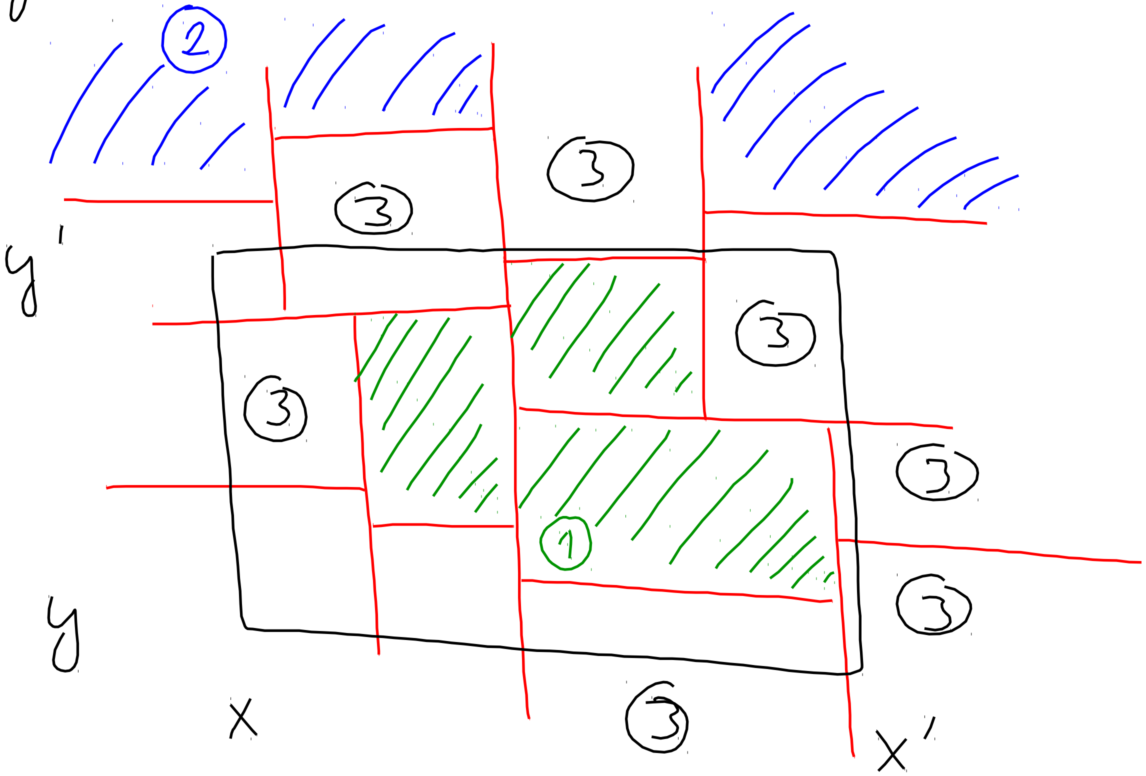
$\text{region}(lc\,v) = \text{region}\ v \cap \text{left}(\ell)$

kde ℓ značí uzel v



$\text{region}(rc\,v) = \text{region}\ v$
$\cap\ \text{right}(\ell)$

Vyhledávání v kd-stromu



Daný region

(1) leží celý v $R = [x, x'] \times [y, y']$

(2) nemá průnik s R

(3) má neprázdný průnik, ale neleží celý v R

(1) všechny body daného regionu leží v R

(2) body regionu neleží v R

(3) musíme dál zkoumat

tomu odpovídá algoritmus

25

Časová náročnost vyhledávání podle kd-stromu v dimenzi 2 je

$$O(\sqrt{n}+k)$$

kde k je počet bodů v R.

Vše se dělá za předpokladu, že žádné 2 body nemají stejnou souřadnici x ani y.

Toto lze odstranit takto:

Místo čísel budeme uvažovat dvojice čísel uspořádané lexikograficky.

Jestliže bod má souřadnice $\bar{x}, \bar{y}$, pak jeho „nové" souřadnice definujeme $(\bar{x},\bar{y})$, $(\bar{y},\bar{x})$

1. souřadnice 2. souřadnice

⑥

Obdĺžnik $R = [x, x'] \times [y, y']$ nahradíme obdĺžnikom

$$R' = [(x, -\infty), (x', \infty)] \times [(y, -\infty), (y', \infty)]$$

1. súradnice 2. súradnice

Platí

$$(\bar{x}, \bar{y}) \in R$$

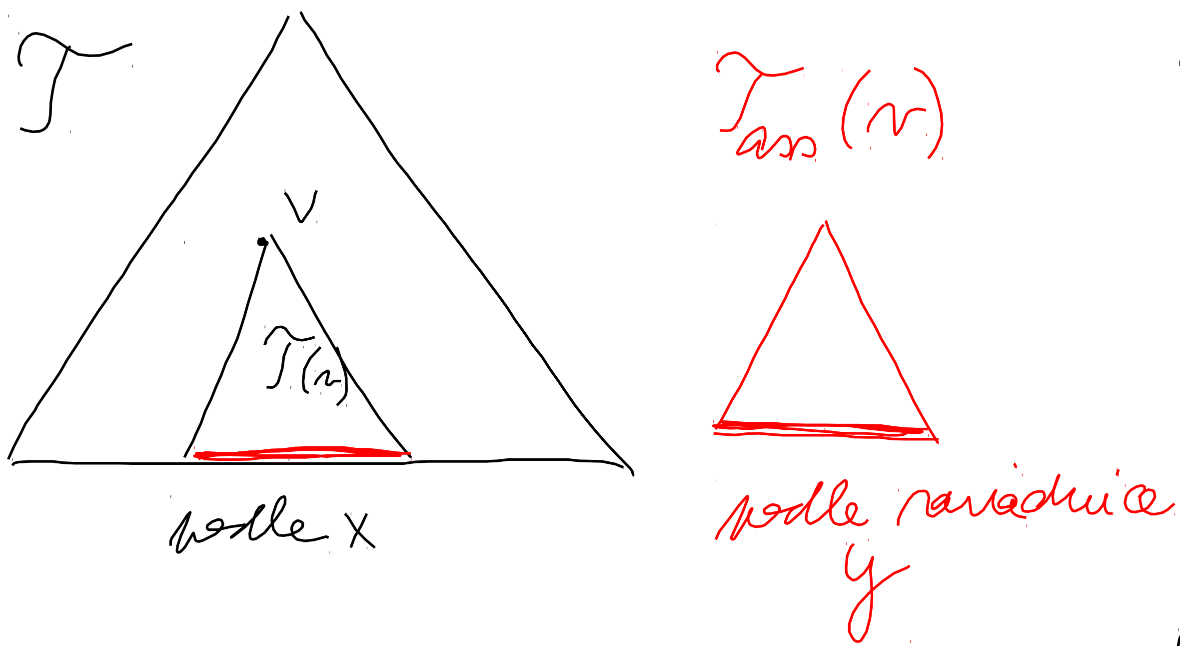
práve keď

$$[(\bar{x}, \bar{y}), (\bar{y}, \bar{x})] \in R'$$

2. metoda – range trees

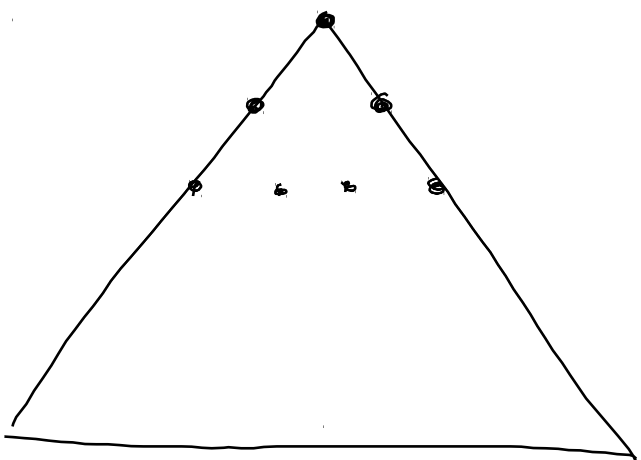
Vyhledávací struktura je následující

Vyvážený binární strom T podle souřadnice x



Ke každému uzlu v v T vytvoříme asociovaný strom $T_{ass}(v)$, který uspořádává množinu bodů v $T(v)$ podle souřadnice y

Nároky na paměť - zatímco v kd-stromu je potřeba $O(n)$ paměti, zde jsou nároky vyšší



$$n = n$$

$$\frac{n}{2} + \frac{n}{2} = n$$

$$\frac{n}{4} + \frac{n}{4} + \frac{n}{4} + \frac{n}{4} = n$$

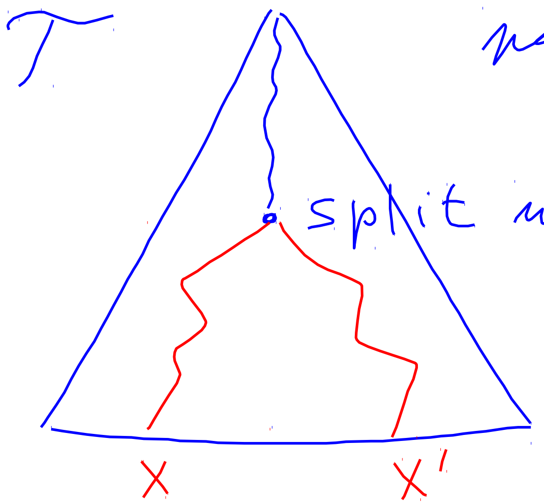
$\log_2 n$

Paměťová náročnost je $O(n \log n)$

Je vyšší, ale vyhledávání bude rychlejší.

Vyhledávání

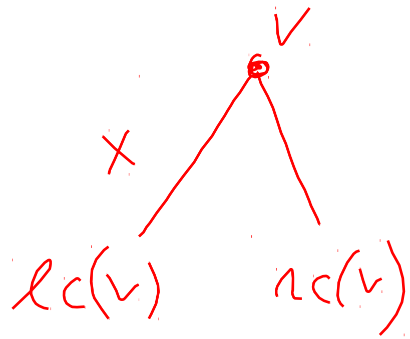
využívá 1-dim vyhledávání



podle x

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

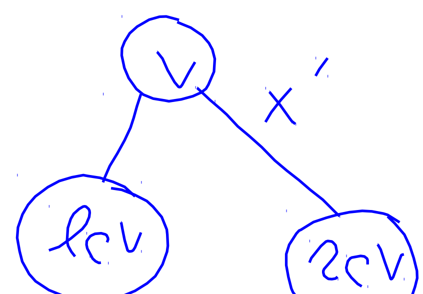
S x jdeme vlevo od v, tak



v $T_{ass}(rc(v))$ vyhledáváme podle y podle 1-dim algoritmu

Obdobně pro cestu x'

Když jdeme z uzlu v vpravo, pak v lc(v) vyhledáváme podle y-souřadnice



Algoritmus 27

(10)

Čas potřebný při vyhledávání v range tree

1D algoritmus potřebuje $O(\log n + k_v)$
v uzlu v

Potřebný čas $\sum_{v} (\log n + k_v)$

v … uzly přes které procházíme

Těch uzlů je řádově výška stromu $\log n$

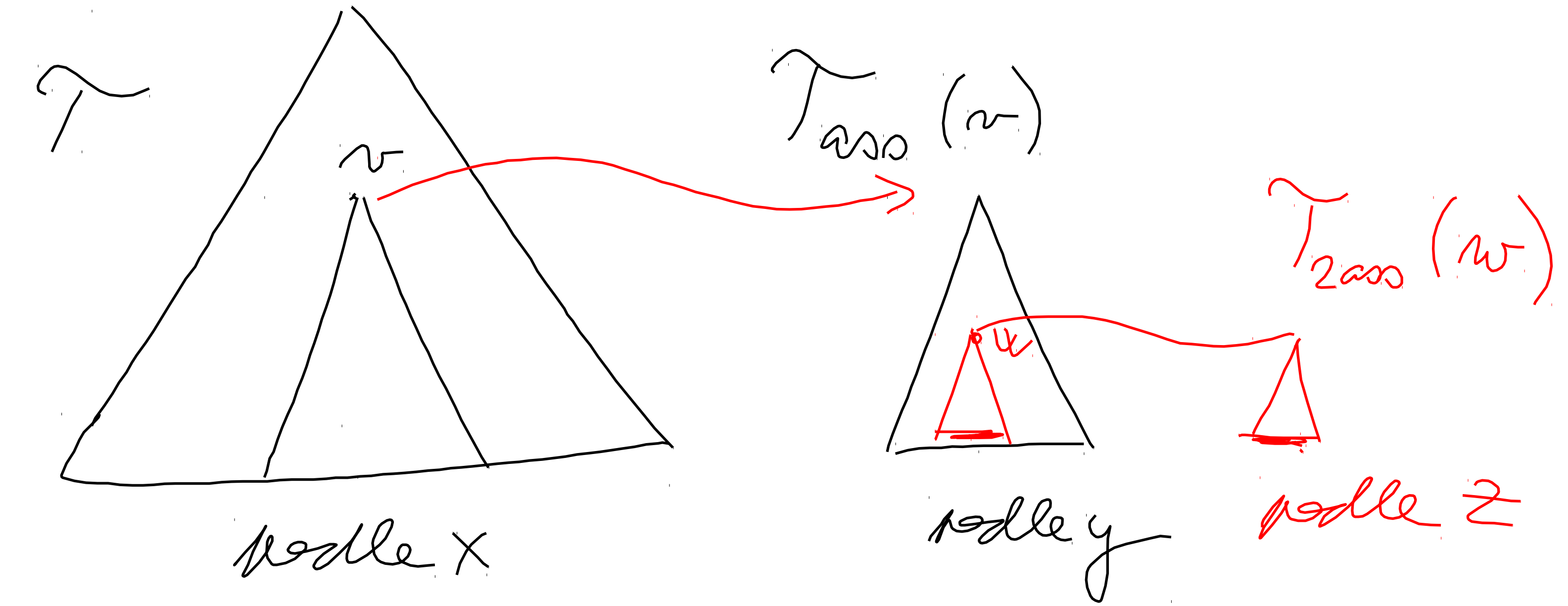
$$\sum_{v} (\log n + k_v) \approx \log n \cdot \log n + \sum k_v = \log^2 n + k$$

Časová náročnost je $O(\log^2 n + k)$

Obě metody jde posádět ve vyšších dimenzích

kd - strom v dim 3 .... vrstvy 3 typů

range tree v dim 3



12

Parametry algoritmů v dimenzi d

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

$d \geq 2$

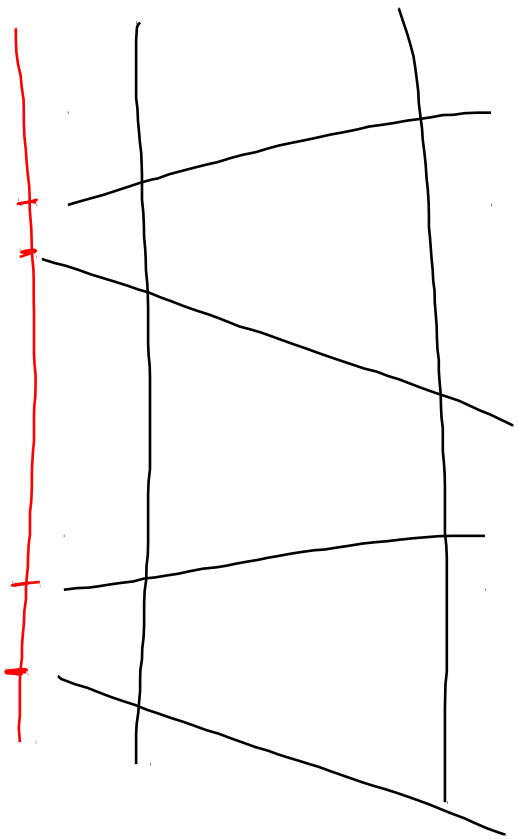
13

LOKALIZACE BODU

Pro rovinné podrozdělení (mapa) chceme najít vyhledávací strukturu, která při zadání souřadnic bodu najde oblast, ve které bod leží.

Vyhledávací struktura se vytvoří pomocí tzv. lichoběžníkové mapy.

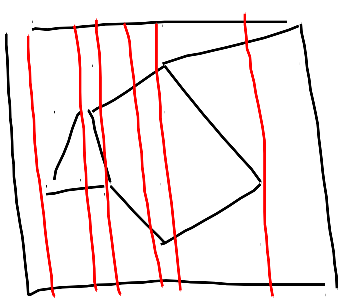
Založeno na tom, že vyhledávání oblasti v „pásu" je jednodušší.



Vyhledávat lze pomocí „nevertikálních" úseček

- nad úsečkou
- pod úsečkou

1. nápad – každým vrcholem vedeme vertikální přímku. Tím mapu rozdělíme na rovnoběžníky.

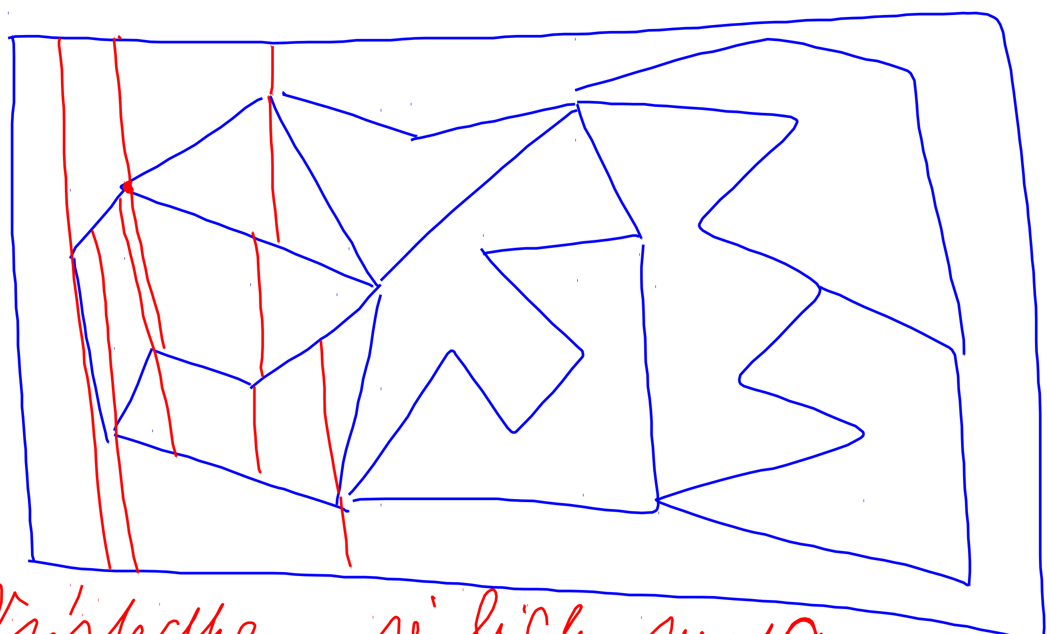


Nevýhoda – z n oblastí můžeme dostat řádově n^2 oblastí – příklad (obr 4) ze kap. 5

Lichoběžníková mapa

Zjednodušující předpoklad – žádné dva vrcholy nemají stejnou souřadnici x.

Rovinné podrozdělení budeme uvažovat uvnitř nějakého obdélníku R

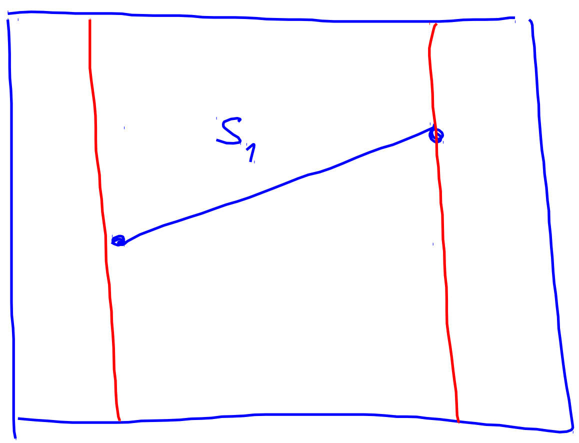


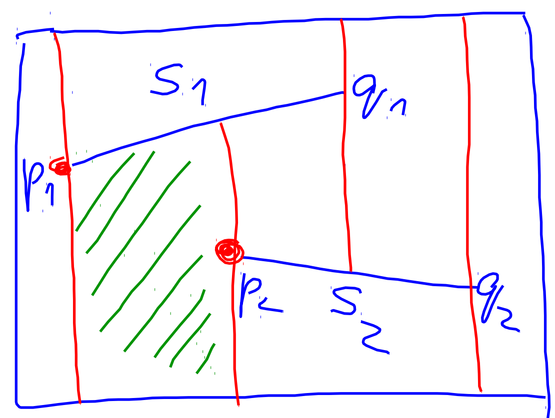
Celé R rozdělíme na lichoběžníky ne na trojúhelníky (deformované lichoběžníky) tak, že z každého vedeme svislou polopřímku nahoru a nejbližší hraně a dělí také k nejbližší hraně.

Výsledkem je lich. mapa.

<u>Popis lichoběžníků</u> Lichoběžníkovou mapu lze vytvořit pro jakoukoliv množinu S n úseček $s_1, s_2, \ldots, s_n$ v obdélníku R

Předpokládáme, že se úsečky neprotínají ve vnitřních bodech.





Zelený lichoběžník je určen

horní hranou s_1
top (úseč)

spodní hrana - dolní hrana R
bottom (úsečka)

pravým vrcholem p_2

levým vrcholem p_1

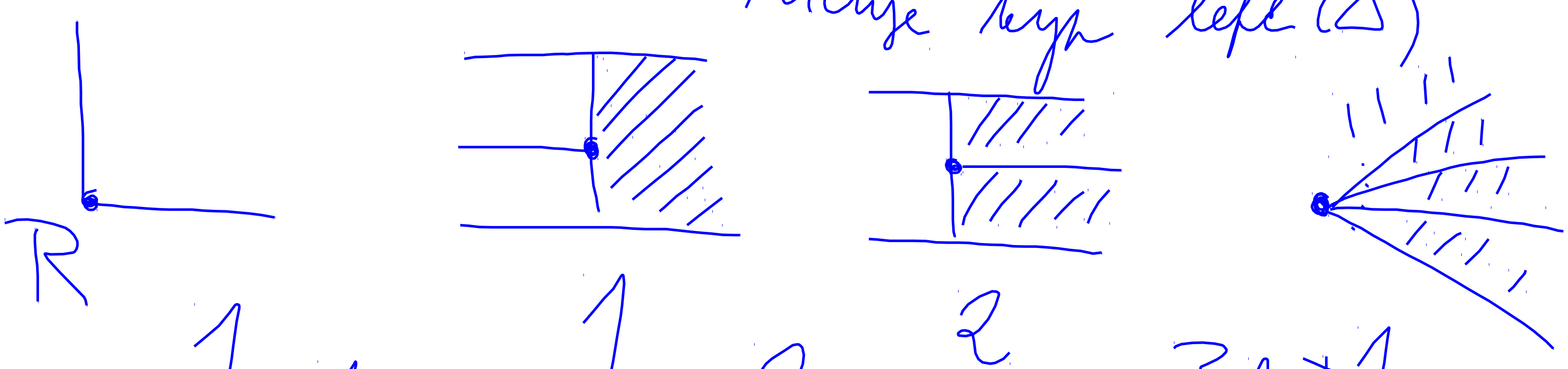
Δ lichoběžník

top(Δ), bottom(Δ), left(Δ)
right(Δ)

Věta: Lichoběžníková mapa pro n úseček má nejvýše $6n+4$ vrcholů a $3n+1$ lichoběžníků.

Dk: Vrcholy = vrcholy $\underbrace{R}_{4}$ + $\underbrace{\text{konc. body úseček}}_{\leq 2n}$ + $\underbrace{\text{nové vrcholy}}_{\leq 2\cdot(2n)}$ $= 6n+4$

Počet lichoběžníků určuje jejich left(Δ)



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