ORTOGONÁLNÍ VYHLEDÁVÁNÍ

Zadána množina P bodů v rovině. Chceme najít strukturu, která umožní rychle vyhledat, které body z P leží v obdélníku $[x, x'] \times [y, y']$, který dostaneme pro dotaz hledání jako vstup.

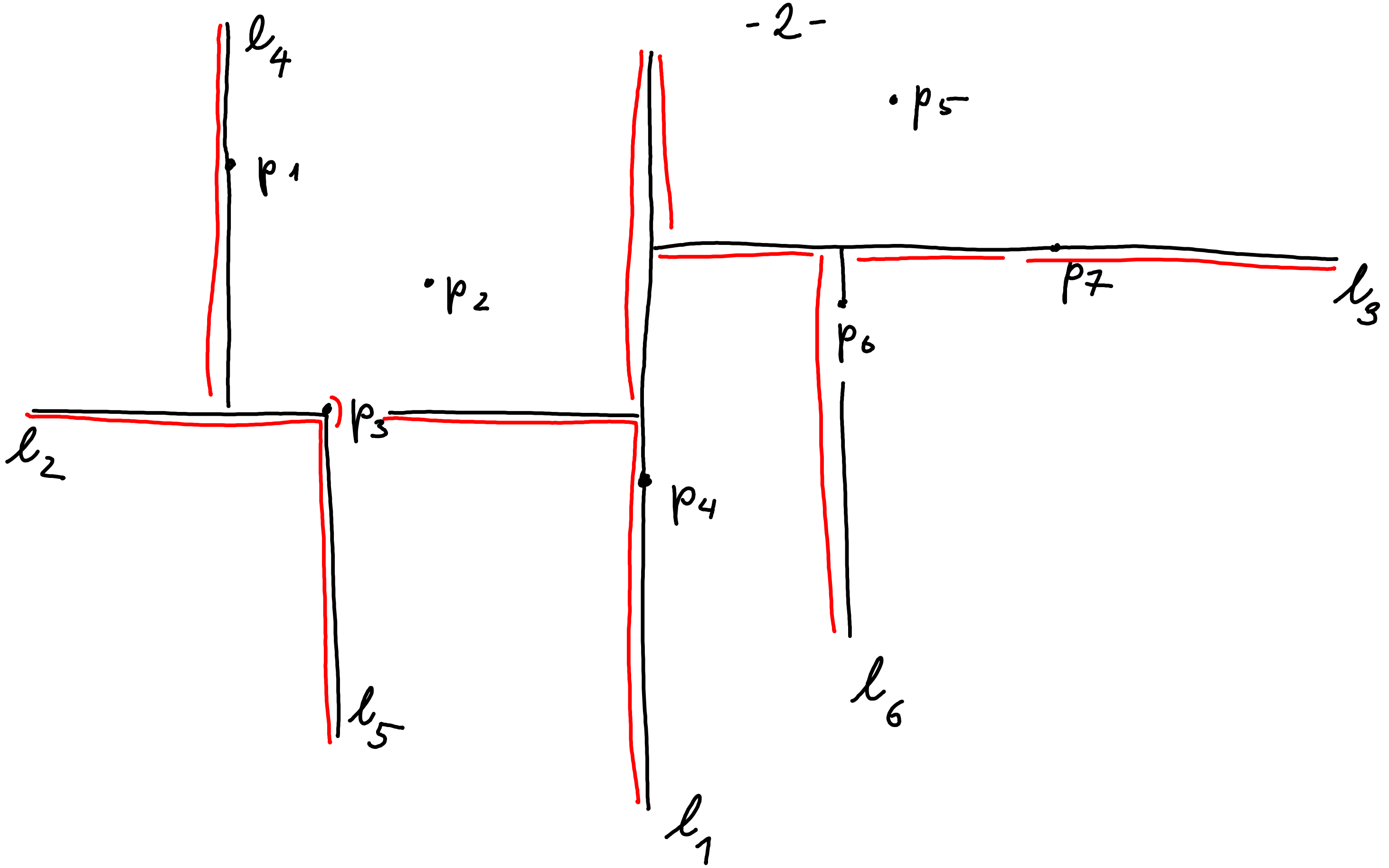
Udělali jsme v dim 1

kd-stromy

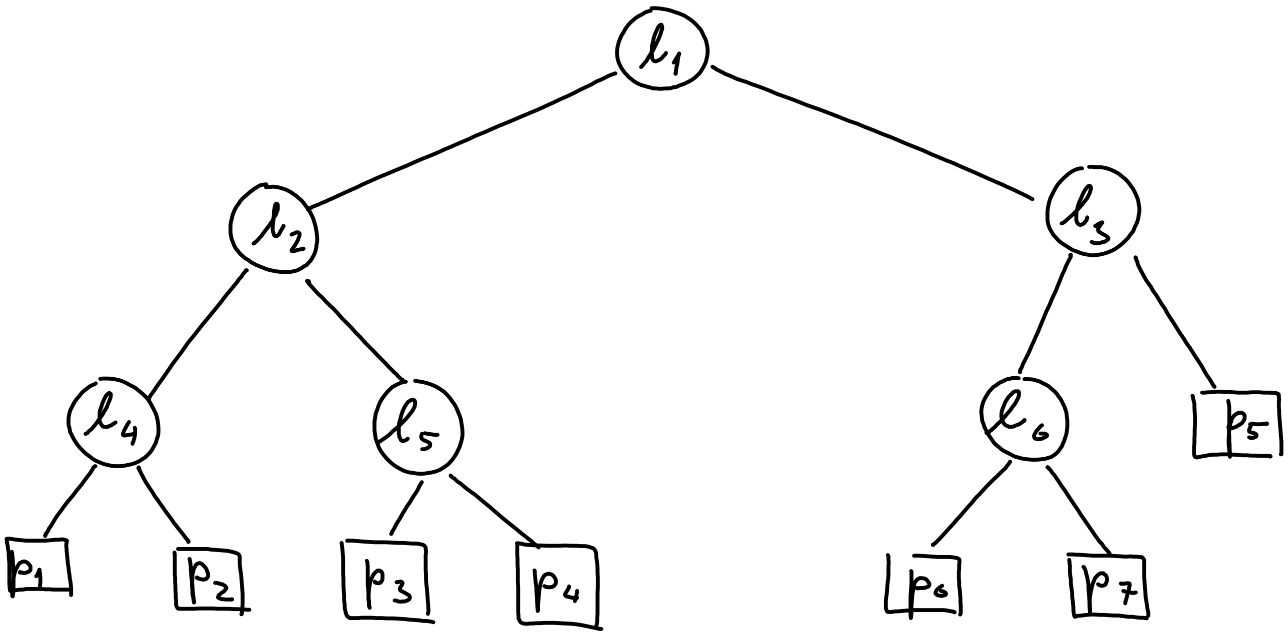
range-stromy

Struktura pro kd-stromy

$P = \{p_1, p_2, \dots p_n\}$ a nichž žádné dva nemají stejnou souřadnici x ani y


-2-
ℓ_4
p_1
p_2
p_5
p_3
p_4
p_6
p_7
ℓ_2
ℓ_3
ℓ_5
ℓ_1
ℓ_6

-3-



Náročnost algoritmu na vyhledávání kd-stromu je $O(n \log n)$.

viz obr 6 a 7, algoritmus 24

-4-

Hledání medianu v zadané množiny.

Množinu P uspořádáme nyní podle x-ové souřadnice

$$O(n \log n)$$

a potom podle y-ové souřadnice.

Nalezení medianu množiny P trvá čas $O(n)$

$P = P_1 \cup P_2$, kde nalezení medianu množiny P_1 trvá čas $O(\frac{n}{2})$

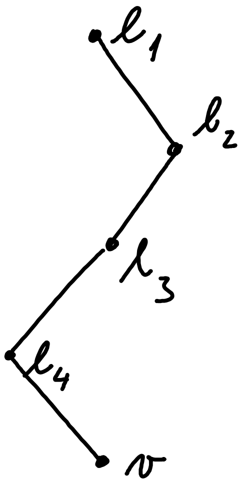
Pro algoritmus 24 pak dostáváme rekurentní formuli pro časovou náročnost

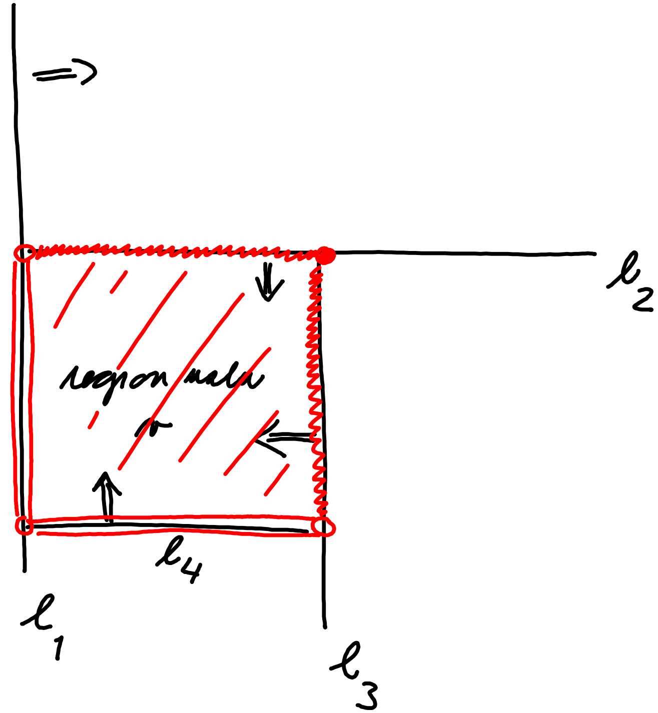
$$T(n) = 2\,T\left(\frac{n}{2}\right) + O(n) \Rightarrow T(n) = O(n \log n)$$

Paměťová náročnost je $O(n)$

-5-

Vyhledávací algoritmus - vhodné definovat pojem **region uzlu v**



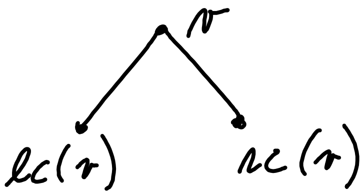


$$\text{region } v = \underbrace{l_1^+ \cap l_2^- \cap l_3^- \cap l_4^+}$$

poloroviny s hranicí (-)
bez hranice (+)

-6-

Rekurentní definice je



$$\text{region } lc(v) = \text{region } v \cap \left(\text{polorovina určená přímkou ve } v\right)^{-}$$

$$\text{region } rc(v) = \text{region } v \cap \left(\text{polorovina určená přímkou ve } v\right)^{+}$$

ALGORITMUS PRO VYHLEDÁVÁNÍ 25

Ilustrativní obrázky 8 a 9.

Časová náročnost: (1) platí rekurentní vztah

$$Q(n) = 2Q\left(\frac{n}{4}\right) + 2 \overset{(2)}{\Longrightarrow} O(n^{\frac{1}{2}} + k) \text{ je časová náročnost}$$

n počet bodů a k je počet bodů v pravoúhelníku

Jak odstranit omezující požadavek na různost souřadnic x a y.

Místo uspořádání v $\mathbb{R}$ pro x-ové a y-souřadnice,

budeme brát lexikografické uspořádání na dvojicích

$p \in \mathbb{R}$ $\quad p = (p_x, p_y)$

Tento bod nahradíme takto

$$\tilde{p} = \left[(p_x, p_y), (p_y, p_x) \right] = \left[\tilde{p}_1, \tilde{p}_2 \right]$$

Pro tyto body uvažujeme v každé složce lexikografické uspořádání

$$\tilde{p}_1 < \tilde{q}_1 \iff p_x < q_x \text{ nebo } p_x = q_x \text{ a } p_y < q_y$$

$$\tilde{p}_2 < \tilde{q}_2 \iff p_y < q_y \text{ nebo } p_y = q_y \text{ a } p_x < q_x$$

-8-

$p \neq q \iff \tilde{p}_1 \neq \tilde{q}_1$ a $\tilde{p}_2 \neq \tilde{q}_2$

Místo intervalů $R = [x, x'] \times [y, y']$

vezmeme

$$\tilde{R} = [(x, -\infty), (x', +\infty)] \times [(y, -\infty), (y, +\infty)]$$

<u>Věta:</u> $p \in R \iff \tilde{p} \in \tilde{R}$

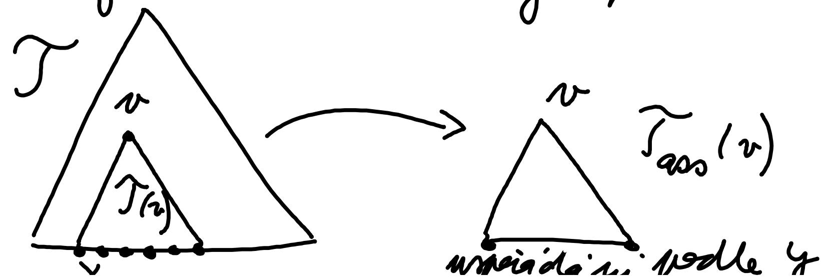
$$x \leq p_x \leq x' \iff (x, -\infty) \leq (p_x, p_y) \leq (x', \infty)$$

Range trees

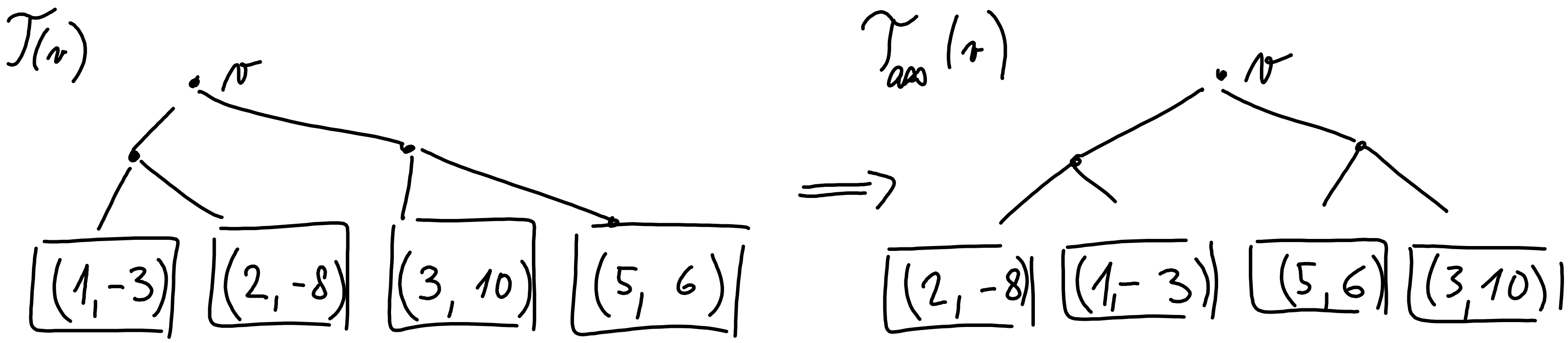
$P = \{p_1, \ldots, p_n\}$, opět předpokládáme, že žádné dva nemají stejnou x-ovou ani y-ovou souřadnici (pokud tomu tak není, vezmeme předchozí lexikografické uspořádání).

T je vyvážený binární strom podle x-ové souřadnice.

v uzel v T vytvoříme pro podstrom s kořenem v asociovaný binární strom s uzlem v, který uspořádává listy podstromu určeného v pomocí souřadnice y.



-10-



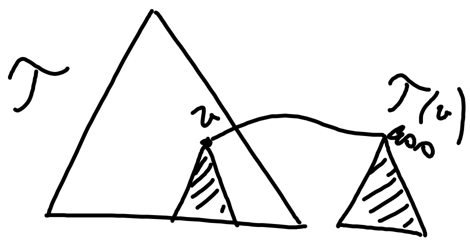
$1 < 2 < 3 < 5$

podle x

$-8 < -3 < 6 < 10$

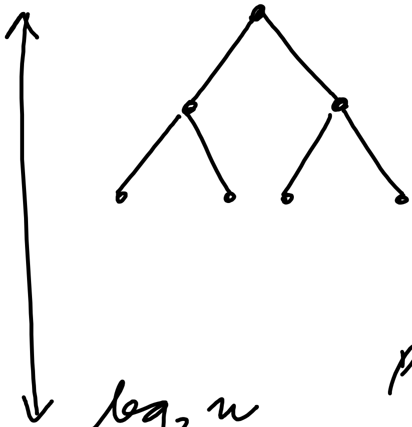
podle y

Vyhledávací struktura = range tree



Paměťová náročnost je $O(n \log n)$

-11-



$\log_2 n$

$$n = n$$

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

$$\frac{n}{4} + \frac{n}{4} + \frac{n}{4} + \frac{n}{2} = n$$

Paměťová náročnost je $O(n \log n)$

$$\log_a n = \frac{\log_b n}{\log_b a}$$

Vyhledávání, které body leží v $R = [x, x'] \times [y, y']$, pomocí range tree – algoritmus č. 27, konstrukce range-tree – alg 26

-12-

Věta. Vyhledávání pomocí algoritmu 27 vyžaduje čas

$$O(\log^2 n + k)$$

kde n je počet bodů v P a k je počet bodů v $R = [x, x'] \times [y, y']$.

Dk. 1 dim. vyhledávání podle y v uzlu v
vyžaduje čas

$$O(\log n + k_v)$$

Celkový čas je nejvýše

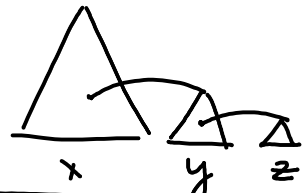
$$\sum_{\substack{v \text{ uzel} \\ \text{kterým procházíme}}} O(\log n + k_v)$$

počet uzlů, kterými procházíme je $O(\log n)$

$$\sum_v k_v = k$$

Tedy

$$\sum_{v} O(\log n + k_v) = \sum_{v} O(\log n) + \underbrace{\sum_{v} k_v}_{k} = O(\log^2 n) + k$$



Porovnání parametrů pro kd-stromy a range-stromy

	kd-strom v dim 2	range tree v dim 2	kd-strom v dim $d \geq 2$	range tree v dim $d \geq 2$
paměť. náročnost	$O(n)$	$O(n \log n)$	$O(n)$	$O(n \log^{d-1} n)$
vytvoření	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n \log^{d-1} n)$
vyhledávání	$O(n^{\frac{1}{2}} + k)$	$O(\log^2 n + k)$	$O(n^{1-\frac{1}{d}} + k)$	$O(\log^d n + k)$

$n = 10\,000$

100

16