Orthogonal searching

P is a finite set in $\mathbb{R}^d$

We want to construct a search structure which enables to find all the points in a prescribed box

$$[x_1, x_1'] \times [x_2, x_2'] \times \dots [x_d, x_d']$$

Two ways – kd trees
– range trees

In dimension 1 both methods coincide

$P \subseteq \mathbb{R}$ $\quad [x_1, x_1']$ We want to find all points in P lying in $[x_1, x_1']$.

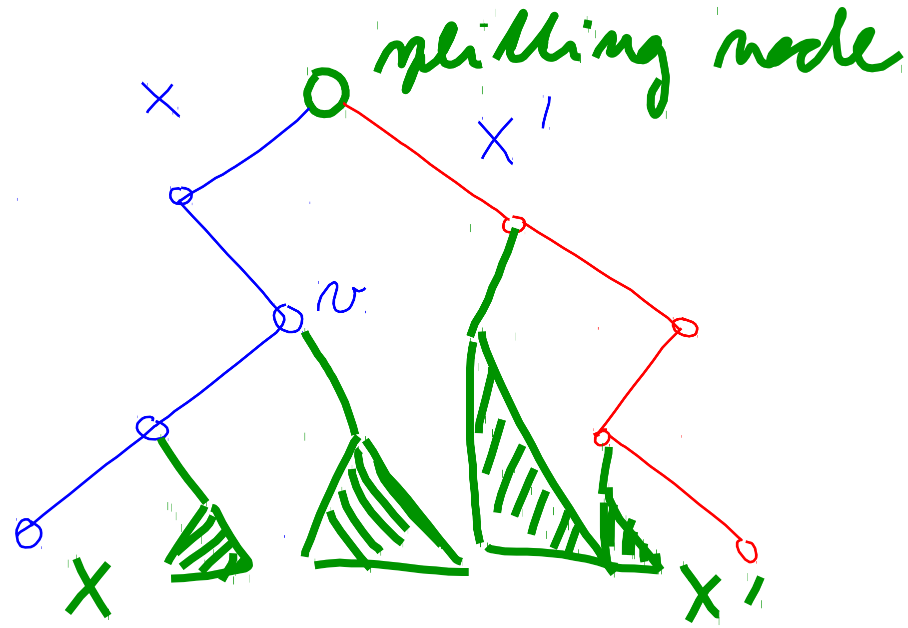
We will use binary trees.

Example $P = \{1, 2, 3, \dots 7\}$

$x_1 = 1{,}5 \quad x_1' = 3{,}5 \qquad x \le x'$

Splitting node for x_1 and x_1' is the last node on the common part of both pathes.

Searching algorithm

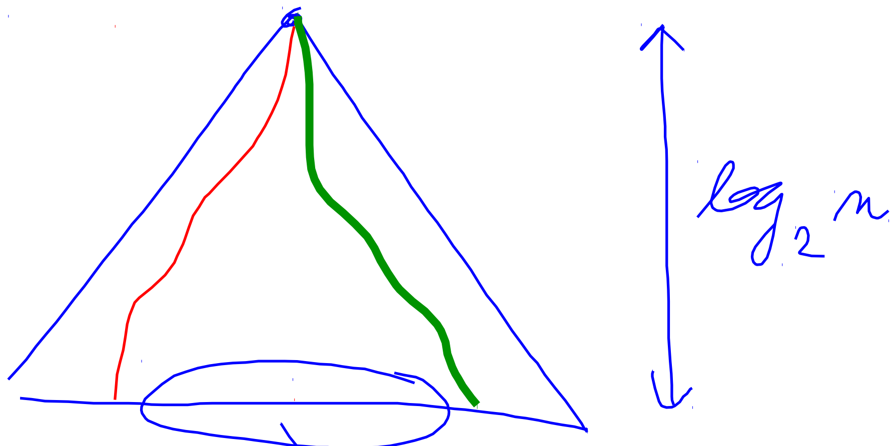


<u>Needs on memory</u>

Binary tree on n leaves

$O(n)$

- running time



There are k points in $[x, x']$

$O(\log n + k)$

kd-trees in dimension 2

Running time for construction of kd-tree

$T(1) = O(1) \qquad T(n) = \underbrace{O(n)} + 2\,T\left(\frac{n}{2}\right)$

Solution of this formula is

$$T(n) = O(n \log n)$$

Searching using kd-tree

Region of a node v

Removing assumption on coordinates

Real $\mathbb{R}^2$ New $\mathbb{R}^2$

(x, y) $((x,y),(y,x))$

$\qquad\qquad\qquad\qquad \neq \qquad \neq$

$(x, z) \quad y \neq z$ $((x,z),(z,x))$

real $\mathbb{R}^2$ new $\mathbb{R}^2$

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

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

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

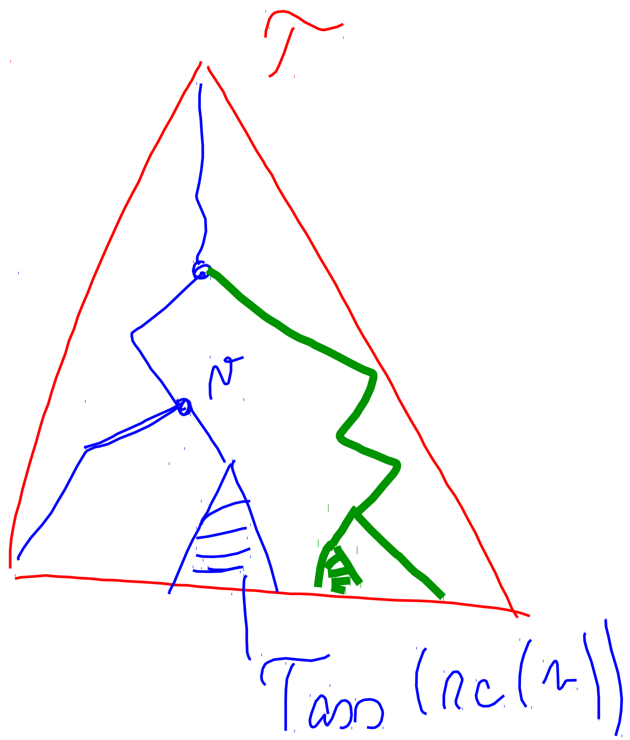
$$\implies \bar{x} \in [x, x'], \quad \bar{y} \in [y, y']$$

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

$\Longleftarrow$

Range trees

- system of a binary tree according to x together with associated ~~sub~~ trees according to y.



Running time

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

$\log^2 n \ll \sqrt{n}$ for big n

x node searching points in v

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

$$\sum_{\substack{v \in \text{path for } x \\ \text{or } x'}} O(\log n_v + k_v) = \sum_{x \text{ path}} O(\log n + k_v)$$

$$= \log n \cdot \log n + k_{v_1} + k_{v_2} + \dots + k_{v_i}$$

$$= \log^2 n + k$$

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

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

$$T(n) = O(n \log^{d-1} n) + 2\,T\left(\frac{n}{2}\right) \quad \ldots\ldots \quad T(n) = O(n \log^d n)$$