L6 27.3.17

Topic: Geometry

static : persistence } ~ automatic
dynamic: retroactivity }

⇝ good for e.g. segments ...

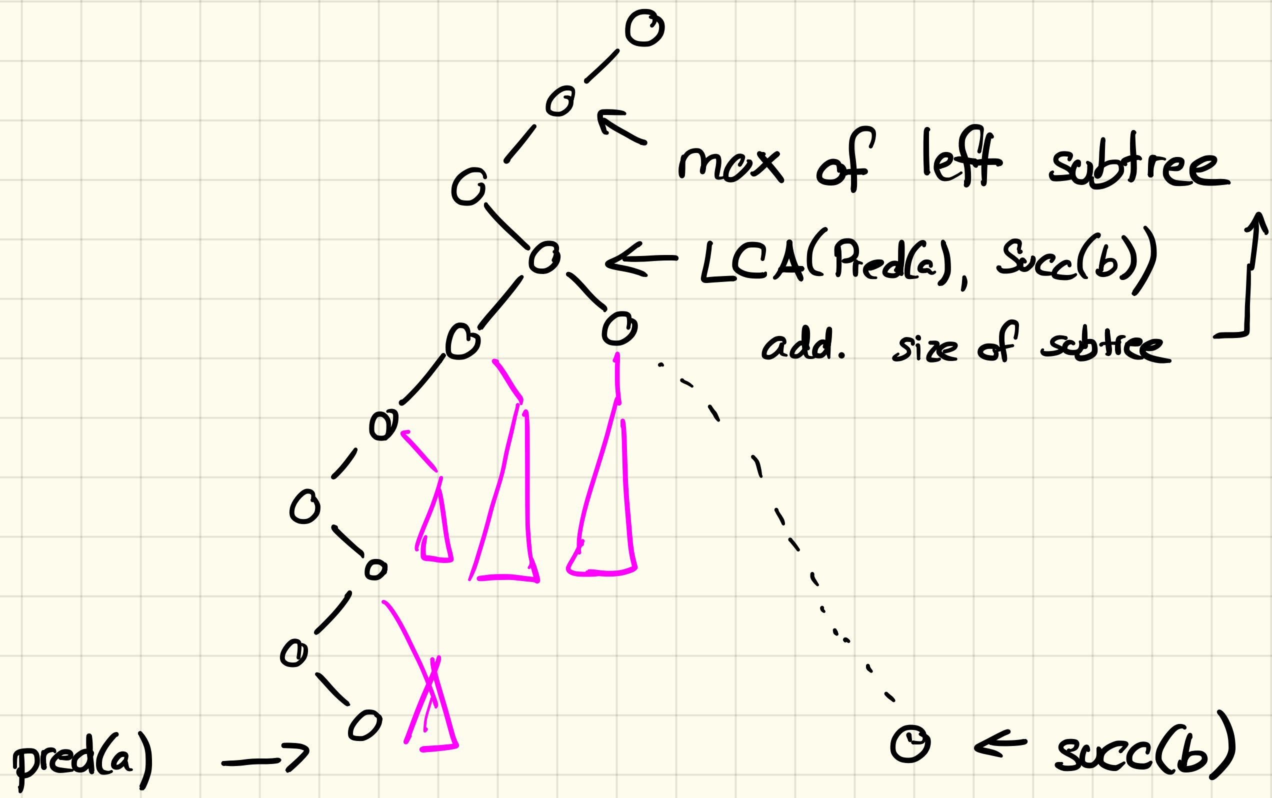
Orthogonal Range Searching

- maintain n points in d dim space
- query: d-dim hyperrectangle
  ↳ $[a_1, b_1] \times \dots \times [a_d, b_d]$
  → existence/count/ list points

- time $\mathcal{O}(\log^d n + k)$

- dynamic/static

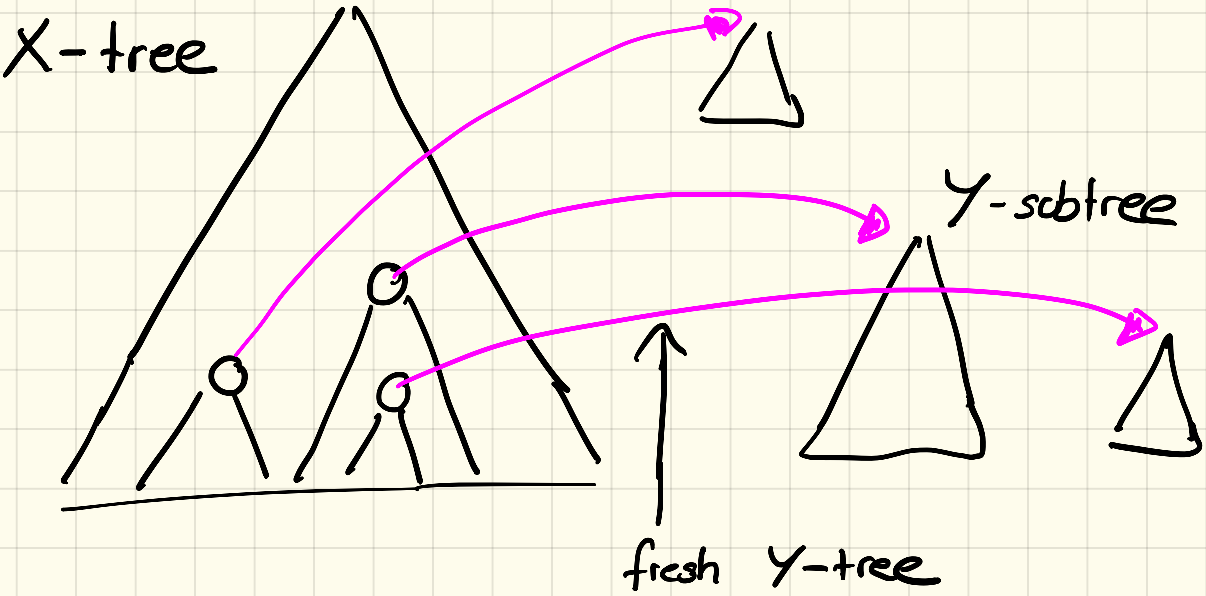
1-dim : Balanced Search Tree



$$\Rightarrow \quad O(\log n + k)$$

2-dim: 1D-tree on X coord

+ each node stores link to 1D tree on Y coord of all its points



- each point is contained in $O(\log n)$ subtrees $\Rightarrow$ $\underline{\Theta(n \log n) \text{ space}}$

- Query$([a_1, b_1] \times [a_2 \times b_2])$:

  Do X-query on $[a_1, b_1]$

  $\rightarrow$ $O(\log n)$ subtrees

  $\forall$ subtrees do Y-query on $[a_2, b_2]$ (on Y-tree!)

  $\Rightarrow$ $\underline{O(\log^2 n + k) \text{ time}}$

exercise

<u>Claim</u>: $\Theta(n \log n)$ construction time.

Updates $\Theta(\log^2 n)$ amortized (for later)

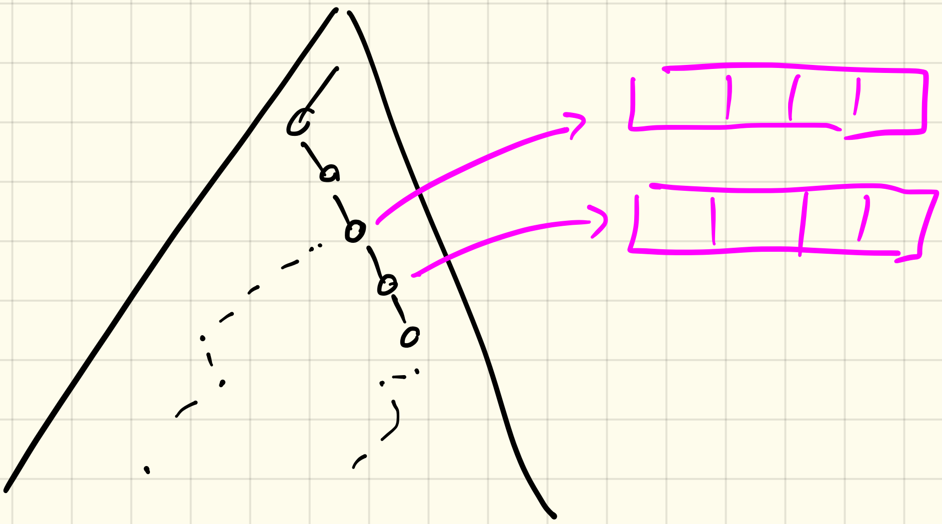
d-dim: Recursion on dimension

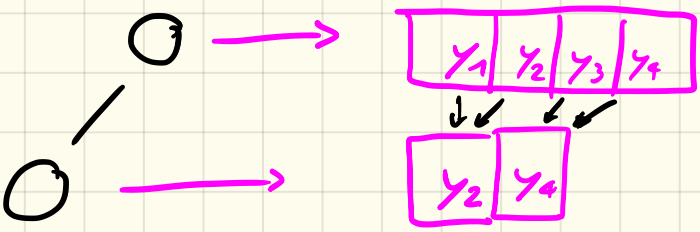
Query: $O(\log^d n + k)$
Space: $O(n \log^{d-1})$
Updates: $\Theta(\log^d n)$ (for later)

Layerd Range Tree (Gabun et. al. 1984)

Idea: Reuse Searches

- Tree on X-coord
- Node stores pointer to sorted array (acc. to Y-coord) of subtree points





- Arrays on Y have pointers to arrays of children

- Query$([a_1, b_1] \times [a_2, b_2])$
  - Binary Search in root for a_2, b_2
  - Follow pointers decomposing X

  $\Longrightarrow \mathcal{O}(\log n + k)$ query

Construction time remains $\mathcal{O}(n \log n)$

d-dim: $\mathcal{O}(\log^{d-1} n + k)$ query
$\mathcal{O}(n \log^{d-1} n)$ construction

Dynamization with Augmentation via weight balance

BB[α] trees

- $\forall$ node x: $size(left(x)) \geq \alpha \cdot size(x)$
  $size(right(x)) \geq \alpha \cdot size(x)$

$\Rightarrow$ height of tree $\leq \log_{\frac{1}{1-\alpha}} n$

- x becomes unbalanced
  $\Rightarrow$ rebuild subtree of x

$\rightarrow$ cost?

Rebuild tree of size k

$\Rightarrow$ Imbalance was $\geq (1-2\alpha)k = \Theta(k)$

pay for rebuilding from Ins / Del in this subtree.

$\Rightarrow$ amortized cost of Ins/Del $\Theta(\log n)$

Dynamic Range Tree:

- Rebuilding of k points costs $\Theta(k \log^{d-1} k)$ (first dim tree)

Change:

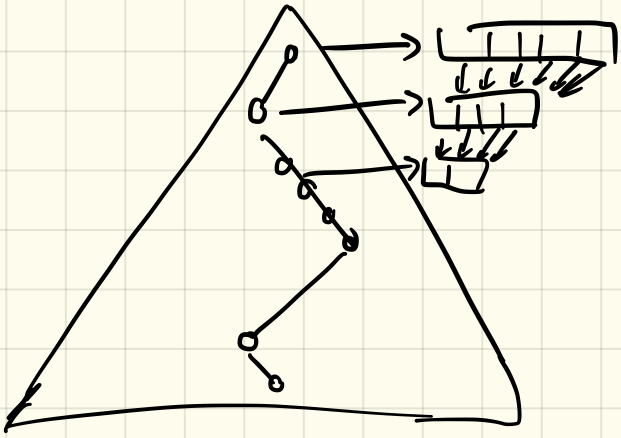
$\Theta(\log n)$ updates à $\Theta(\log^{d-1} n)$ on 1^{st} dimension.

$\Theta(\log^2 n)$ updates à $\Theta(\log^{d-2} n)$ on 2^{nd} dim

. . .

$\Longrightarrow$ total $\Theta(\log^d n)$

Dynamic Layered Range Tree



- arrays → linked lists
- root array → BST

$\Longrightarrow$ Update: $\Theta(\log^d n)$

Query: $\Theta(\log^{d-1} n + k)$