Let $L(u)$ = original y-sorted list at node u

Will generate an "augmented list" $L^+(u)$

Define $\text{sample}(L)$ = sublist of L formed by taking 1 out of every b elem

for each child v of u,

let $L^+(v) = \boxed{L(v)} \cup \text{sample}(L^+(u))$

store succ ptrs between $L^+(v)$ and $\text{sample}(L^+(u))$

& succ ptrs between $L^+(v)$ and $L(v)$

if we know succ of q in $L^+(v)$,

know succ of q in $L(v)$. ✓

know succ of q in $\text{sample}(L^+(u))$

$\Rightarrow$ know succ of q in $L^+(u)$

by additional $O(b)$ comparisons

$O(1)$ time

repeat at u

$\Rightarrow$ query time $O(\log n + (\log n) \cdot O(1))$

init binary search at leaf

$= \boxed{O(\log n)}$

Space

$$O\Big(\sum_v |L^+(v)|\Big) = O\Big(\sum_u |L(u)| \Big(1 + \frac{2}{b} + \Big(\frac{2}{b}\Big)^2 + \dots\Big)\Big)$$

$b > 2$

$$= O\Big(\sum |L(u)|\Big)$$

$b > 2$ $\quad = O\left(\sum_u |L(u)|\right)$

$= \boxed{O(n \log n)}$

Method 3: Persistent Search Tree (Sarnak–Tarjan '86)

go back to slab Method ()

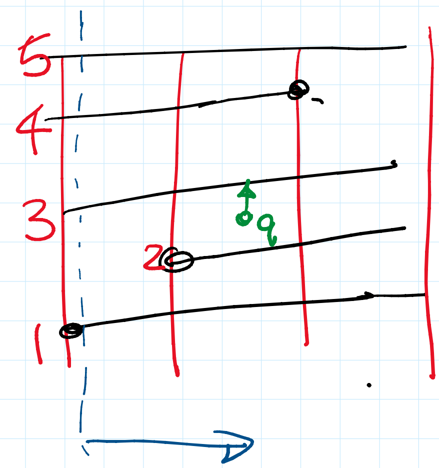
sweep from left to right

maintain y-sorted list L

if we hit left endpt
 insert to L

if right endpt
 delete from L

Store L in BST



to answer query for pt q:

find slab σ containing by x-binary search $O(\log n)$

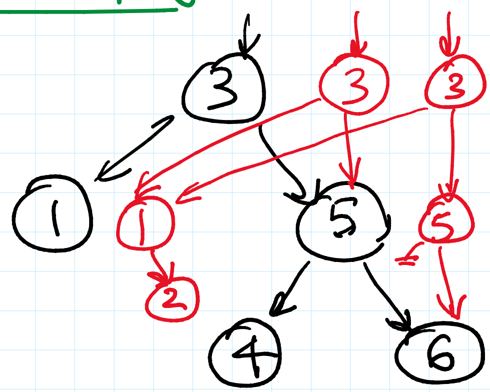
do pred search in the version of L for σ

persistent data structure - ability to query the past versions.

One implementation of persistent BST:

One implementation of persistent BST:

path copying



insert(2)
delete(4)

query time $O(\log n + \log n) = \boxed{O(\log n)}$

bin search, in x

search time for BST.

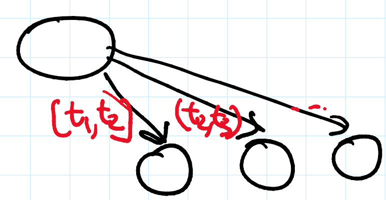
Space $\boxed{O(n \log n)}$

we create $O(\log n)$ new nodes per insert/delete

2^{nd} implementation: no path copying

allow node to get "fat"

store timestamps for ptrs



space $\boxed{O(n)}$

if we use BST with $O(1)$ amort. update time

query time $O(\log n + \log n \cdot \log n)$

$= \boxed{O(\log^2 n)}$

or with vEB trees

$O(\log n + \log n \cdot \log\log n)$
$= \boxed{O(\log n \log\log n)}$

Sarnak-Tarjan: limited path copying

allow node to get "fat" with up to b slots
copy when full

$\Rightarrow$ query $\boxed{O(\log n)}$
space $\boxed{O(n)}$ by some potential analysis
(even for $b=1$!)