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 \( b \) elements.

For each child \( v \) of \( u \),
let \( L^+(v) = \text{sample}(L^+(u)) \)
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.

Repeat at \( u \)

\[ \text{query time } O(\log n + (\log n) \cdot O(1)) = O(\log n) \]

\[ \text{Space } O\left( \sum_{u \in L^+(u)} L(u) \right) = O\left( \sum_{u \in L^+(u)} \left( 1 + \frac{2}{b} + \left( \frac{2}{b} \right)^2 + \ldots \right) \right) \]

\[ = O\left( \sum \left| L(u) \right| \right) \]
\[ b^2 = \mathcal{O} \left( \sum_u |L(u)| \right) = \mathcal{O}(n \log n) \]

---

Method 3: Persistent Search Tree (Samak-Tarjan '86)

- Go back to Method 1
- Slab 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 \( \sigma \) containing \( q \) by x-binary search
- Do pred search in the version of \( L \) for \( \sigma \)

Persistent data structure - ability to query in past versions.

One implementation of persistent BST:
One implementation of persistent BST:

Path copying:

Query time: $O(n \log n)$

Space: $O(n \log n)$

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

2nd implementation: no path copying

Allow node to get "fat"
Store timestamps for ptrs

Space: $O(n)$

If we use BST with $O(1)$ amortized update time,

Query time: $O(n \log n + \log n \cdot \log n) = O(n \log^2 n)$

Or with VEB trees
Sarnak–Tarjan: limited path copying

allow node to get ‘full’ with up to \( b \) slots

\[ \text{copy when full} \]

\[ \Rightarrow \text{query } O(\log n) \]

\[ \text{Space } O(n) \]

by some potential analysis

(even for \( b=1 \))