BST - ordered dictionary

rotation

double rotations

Splay(x)

while parent(x) ≠ root and x ≠ root
  double rotate (x)
if x ≠ root
  rotate(x)

Time for
  find, ins, pred, del
= O(splay) = O(depth(x))

Amortized time: starting with empty/balanced
Theorem: Splay executes any sequence of N
  splays in O(N \log n) time

Proof: potential method

size(v) = # descendants of v
rank(v) = \lceil \log \text{size}(v) \rceil
\Phi(T) = \sum rank(v)
Splay Trees

Amortized time (op) := Time (op) + \Phi_{new} - \Phi_{old} \\
\sum_{op} AT (op) = \sum_{op} T (op) + \Phi_{final} - \Phi_{init} \\
\sum_{op} T (op) \leq \sum_{op} AT (op)

Access Lemma: [Sleator-Tarjan 85]

Am. time to rotate v \leq 1 + 3 \text{rank}' (v) - 3 \text{rank} (v) \\
Am. time to dbl-rot v \leq 3 \text{rank}' (v) - 3 \text{rank} (v) \\
\Rightarrow Am. time to splay v \leq 1 + 3 \text{rank}_{final} (v) - 3 \text{rank}_{init} (v) \\
\leq 1 + 3 \text{rank}_{final} (v) \\
\leq 1 + 3 \log n

Suppose we search for x t (x) times \\
T = \sum_{x \leq n} t (x)

Access Lemma still works if size (v) = \sum_{x \leq v} t (x) \\
\text{rank} (v) = \log \text{size} \\
\Phi = \sum_{v} \text{rank} \\
\Rightarrow Am. time to splay x = 1 + 3 \cdot \log T - 3 t (x) \\
= 1 + 3 \left( \log \frac{T}{t (x)} \right) \quad \text{(Static optimality)}

Conjecture: splay trees are dynamically opt

O(1)-competitive ratio vs. best dynamic BST offline
Dynamic BST?

\[ \mathcal{P} = \text{some upward-closed subset of nodes includes } x \]

arbitrarily reconfigure \( \mathcal{P} \) in \( O(|\mathcal{P}|) \) time

Weaker Conjecture: There is an O(1)-competitive dynamic BST

Geometric View of BSTs:

Access sequence \( x_1, x_2, \ldots, x_n \Rightarrow (x_i, i) \)

Execution sequence = \[ \{ (y, i) \mid \text{node } y \text{ is touched during } i\text{th access} \} \]

Thm: Execution pts are satisfied meaning every rect defined by two pts has another pt on its boundary

Thm: And vice versa.

BAD NEWS: Finding \( \min \) satisfying superset is NP-hard (if we allow multi-accesses)
Natural Greedy heuristic:

Greedy Future:

\[ \text{For } i = 1 \text{ to } N \]
\[ \text{add min #pts on row } i \]
\[ \text{to satisfy rectangles with bottom row } i \]

\[ \mathcal{O}(n \log n) \text{ time} \]

\text{Conj. Greedy Future is } \mathcal{O}(1)-competitive

BST language — optimally reconfigure only search path

\text{Greedy Post is an online algorithm} \Rightarrow \text{Greedy BST}

\[
\text{Greedy } \geq \text{OPT } \geq \max \{ \text{Greedy}, \text{greedy} \bar{R} \}
\]

\text{Conj.} \quad \text{Greedy } \leq \text{OPT } + \mathcal{O}(1)