Data Structure  Dictionary

S is an ordered set.
1) \( \text{Search}(k, S) \equiv x \in S? \)
2) \( \text{Insert}(k, S) \equiv \)
3) \( \text{Delete}(k, S) \equiv \)

Note: If 1, 2, 3) are the Design Requirement
then use a Hash Table

4) \( \text{Range}(k, k', S) \equiv \{ k'' \in S \mid k \leq k'' \leq k' \} \)

If 1), ..., 4) use BST
**Def** A tree $T$ in BST for keys $S$ if:

1) $T$ is an ordered binary tree with 15 nodes.
2) Each node stores a key.
3) Keys are in inorder

$$S = \{a, b, c, d, e\}$$

$$T = \begin{array}{c}
\text{bst}
\end{array}$$

**T is balanced if** $\text{max depth}(T) = O(\log n)$
Types of Balanced BST's

Always Balanced - AVL, 2-3-4, RB, B-Trees
Randomized - Skip-list, Treaps = tree-heaps
Amortized - Splay Trees

All these use the Rotation

To show: inorder is preserved
Applications

Persistence (undo)

First Idea  Keep a tree for each time

Time

$T_1$  $T_2$  $T_n$

$O(n^2)$ space.

Idea 2  Store only differences  Insert $(x)$

$O(\log n)$ space per new tree.
Vanilla - Insert \((k, T)\)
1) Find empty leaf node
2) add \(k\) to leaf

\[ QS(A) \quad \text{Bad Quick Sort} \]
1) Pick first \(a\) in \(A\)
2) Split \(S\) into \(S < a\), \(a\), \(a > L\)
3) Return \(QS(S), a, QS(L)\)

Vanilla - BST \((A, T)\) VBST
1) Extract first \(a\) from \(A\)
2) Return \(VBST(A, VI(a, T))\)

Note: \(QS\) & V-BST do exactly the same comparisons but in different order.
TREAPS

Keys ∈ 1, ..., n
Priorities ∈ p(k), p(k) ≠ p(l) for k ≠ l

T = tree with a key at each node.

Def: T is in heap order if ∀x ∈ T x ≠ root

p(parent(x)) < p(x)

Lemma: A heap order BST exists and is unique.

Pf: Vanilla-insert in priority order.
Random Treap

Insert $(k) \equiv$
1) insert $k$ into a leaf
2) pick random $p(k)$
3) rotate $k$ up until in heap order

Delete $(k) \equiv$
1) rotate $k$ to a leaf by picking highest priority child
2) remove $k$.

Correctness:
Expected Cost for Treaps

**Goal:** Determine expected # of comparisons to search \( (m, 5, K) = S(m, 5) \)

**eg:** \( K = \{1, 2, 3\} \) & search \( (2.5, K) \)

Treaps | Insert order
--- | ---
(123) | (132) | (213) | (231) | (312) | (321)

# | 3 | 3 | 2 | 2 | 3 | 2

\[
S(2.5) = \frac{15}{6} = 2.5
\]

Note for random BST \( \frac{13}{5} = 2.6 \)
Keys $\{1, \ldots, n^3\}$ in Treap

$C(i,m) = \text{Event } [\text{i is compared to m in Search(m)}]$

Claim $\text{Prob}[C(i,m,5)] = \frac{1}{m-i+1}$ if $i \leq m$

$= \frac{1}{i-m}$ if $i > m$

Treap construction as dart game

In informal arguments, WLOG assume $\text{Prob}[C(i,m,5)] = \frac{1}{m-i+1}$
**Formal Argument**

\[ B_j = \text{Event} \left[ j^\text{th} \text{ clot first to land in } [i, \ldots, m] \right] \]

\[ B_k \cap B_j = \emptyset \text{ for } k \neq j \quad \Pr \left( \bigcup_{j=1}^{\infty} B_j \right) = 1 \quad (\star) \]

**Conditional Probability**: Let \( A, B \) be events

\[ \Pr \left[ A \mid B \right] = \frac{\Pr \left[ A \cup B \right]}{\Pr \left[ B \right]} \]

\[ \Pr \left[ C(i,m,z) \right] = \sum_{j=1}^{\infty} \Pr \left[ B_j \right] \Pr \left[ A \mid B_j \right] \quad \text{true by } (\star) \]

Note: \[ \Pr \left[ A \mid B_j \right] = \frac{1}{m-i+1} \]

\[ \Pr \left[ A \right] = \sum_{j} \Pr \left[ B_j \right] \left( \frac{1}{m-i+1} \right) = \frac{1}{m-i+1} \sum_{j=1}^{\infty} \Pr \left[ B_j \right] = \frac{1}{m-i+1} \]

QED
\[ S(m,s) = \# \text{comparisons search } m, s \]

\[ = \sum_{i \neq m} C(i, m, s) \]

\[ E(S(m,s)) = \sum E(C(i,m,s)) = E(\text{Prob}(C(i,m,s))) \]

\[ = \sum \frac{1}{n-i+1} + \sum_{i \neq m} \left( \sum_{i=1}^{n} \frac{1}{i} \right) = 2 \sum_{i=1}^{n} \frac{1}{i} \]

\[ = 2 \ln n + O(1) \]

**Expect # of Comparisons for**

**Insert, Search, Delete = O(\log n)**
Counting Rotations

2 Cases: Insert & Delete

Delete: Move-to-Leaf

Thm: Expect #rotations < 2

Claim: Possible pivots with m are

1) Right-most nodes in left subtree of m
2) Left" "right"

\[ \text{Diagram of tree with pivots marked} \]
Def

\(D_i = \text{Event}[i \text{ is a rightmost node in left subtree of } m]\)

\(D_i' = \text{Event}[i\text{'s left'} \text{'s right'}\text{ '}]\)

Claim \(P_e[D_i] = \left(\frac{1}{m-i+1}\right)\left(\frac{1}{m-i}\right)\)

pf Dart Game (Informal)

Needed so that \(D_i:\)

\[
\begin{array}{c|c|c|c|c}
1 & i & m & n \\
\hline
\text{don't care} & \uparrow & \text{remaining darts} & \uparrow & \text{don't care} \\
\text{second} & & & & \text{first dart}
\end{array}
\]

\(P_e(D_i) = \left(\frac{1}{m-i+1}\right)\left(\frac{1}{m-i}\right)\)

\(P_e(D_i') = \left(\frac{1}{i-m+1}\right)\left(\frac{1}{i-m}\right)\)
Def \( R_m = \# \text{rotations in move-to-leaf of } m \)

\[
E(R_m) = \sum_{i < m} E(D_i) + \sum_{i > m} E(D'_i)
\]

\[
\leq \sum_{i=0}^{m-1} \frac{1}{(m-i+1)(m-i)} + \sum_{i=1}^{n-m} \frac{1}{(i+m)(i-m)}
\]

\[
\leq \sum_{i=1}^{m-1} \frac{1}{(i+1)i} + \sum_{i=1}^{n-m} \frac{1}{(i+1)i}
\]

\text{not}

\[
\frac{1}{(i+1)i} = \frac{1}{i} - \frac{1}{i+1}
\]

\text{hence}

\[
\sum_{i=1}^{m-1} \frac{1}{i} - \sum_{i=2}^{m} \frac{1}{i} = 1 - \frac{1}{m} < 1
\]

\[
E(R_m) < 2
\]