Treap Analysis

Traverse across height of tree - O(height) → O(log n)

Proving Expected height

Let x_k → Node with the k^{th} smallest key in the treap.

S2: To prove that for any node x_k.

S3: $E[\text{depth}(x_k)] = O(\log n)$.

Let y_{ij} = Indicator Random Variable defined as I.

$y_{ij} = 0$; when x_i is not a proper ancestor.

$y_{ij} = 1$; when x_i is a proper ancestor.

$y_{ii} = 0$; Can not be a proper ancestor.

Hence depth → $(x_k) = \sum_{i=1}^{n} y_{ik}$

Expected Depth of x_k is equal to Expected number of proper ancestors of x_k.

$$E[\text{depth}(x_k)] = E\left[\sum_{i=1}^{n} y_{ik}\right] = \sum_{i=1}^{n} E[y_{ik}]$$

$$E[\text{depth}(x_k)] = \sum_{i=1}^{n} Pr[y_{ik} = 1]$$

Deletion in Treaps

To delete:

: Search Node X containing k using BST algorithm.

: If Node X is a leaf; just delete it.

: If Not, use AVL rotations to rotate the node until it becomes leaf; then delete it.

: If 2 child → Rotate Child with larger Priority;
(using Left Rotation if Right child has higher Priority)

: Since AVL rotations are Const-time operations, delete in treap can be performed in time O(H); where H is the height of treap.

Trees Splitting:

Given a tree and a key value k not in tree, create two trees: One with keys less than k, and one with keys greater than k.

- Insert (k, ∞) in the treap.
- Since, it has higher priority; it will become Root.
- Left Subtree of Root, has lower value than k., the right Subtree of Root, has greater value than k.
- Since insert can be done in time O(H), where H is height of treap, splitting can also be done in time O(H)
- Trees Splitting Mechanism,,

Cost of Split $\propto$ Depth of the node.

Expected depth of node = $O(\log n)$

Expected Cost of Split = $O(\log n)$

For Join (T_1, m, T_2)

Follows down the right Child and terminates, when right Child is Empty

T_2

Hence:

The work is proportional to the sum of depth of the right-most and left-most keys.

The work of JOIN = Sum of Expected depth of nodes

$= O(\log|T|)$.

Insert, update, search

Randomised search tree mathematical analysis.

N nodes $x_1, x_2, \ldots, x_N$
keys → $k_1, k_2, \ldots, k_N$
priorities → $p_1, p_2, \ldots, p_N$

x_i node has key k_i and priority p_i

Making 2 assumptions.

1) All the keys $k_1, \ldots, k_N$ are equally likely to be searched for.

2) All the priorities are $p_1, \ldots, p_N$ are randomly uniformly generated independent of each other and of the keys.

Also assuming keys are in sorted order
$k_i < k_{i+1}$

1) But the keys can be inserted in any order.
2) All the priorities are distinct.

Expected node depth

Depth of node $x_i \rightarrow d(x_i)$
This means that $d(x_i)$ comparisions are required to find the key k_i

$Pr(p_1, \ldots, p_N)$ → probability of generating N probrity values
→ This will determine shape of the heap.
and also location of every key in it.
This helps to determine depth of node x_i

$$E[d(x_i)] = \sum_{p_1,\ldots,p_N} Pr(p_1,\ldots,p_N)\, d(x_i) \quad -(1)$$

let A_{ij} be the indicator function.

$A_{ij} = 1$, if x_i is ancestor of x_j

$$\therefore d(x_i) = \sum_{m=1}^{N} A_{mi} \quad \text{(As depth of node is equal to number of ancestors).} \quad -(2)$$

Using (2) in (1)

$$E[d(x_i)] = \sum_{p_1,\ldots,p_N} Pr(p_1,\ldots,p_N) \sum_{m=1}^{N} A_{mi}$$

$$= \sum_{m=1}^{N} E[A_{mi}]$$

$(E[A_{mi}] = Pr(A_{mi} = 1))$ → Probability that node x_m is ancestor of x_i

∴ The probability that x_m is ancestor of x_i is just the probability that the random priority generated for x_m is higher than the other $|m-i|$ priorities generated for nodes with indexes between m and i inclusive.

But priorities are generated randomly and independently $|m-i|+1$ nodes have equal probability of having the highest priority.

$$\therefore E[A_{mi}] = \frac{1}{|m-i|+1}$$

$$\therefore E[d(x_i)] = \sum_{m=1}^{N} \frac{1}{|m-i|+1}$$

$$= \sum_{m=1}^{i-1} \frac{1}{i-m+1} + \sum_{m=i+1}^{n} \frac{1}{m-i+1}$$

$$\left(H_n = \sum_{i=1}^{n} \frac{1}{n}\right) \quad = H_i - 1 + H_{n-i+1} - 1$$

$$< 2 \ln n$$

$$= O(\log n)$$

Average number of comparisions needed to find the key.

This will be equal to expected node depth averaged over all nodes.

$$D_{avg}(N) = \sum_{i=1}^{N} \frac{1}{N} \left(E[d(x_i)]\right) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{1}{|j-i|+1}$$

$$= \frac{2}{N} \sum_{i=1}^{N} \frac{N-i+1}{i} - N$$

$$= \frac{2(N+1)}{N} \sum_{i=1}^{N} \frac{1}{i} - 3N$$

$$= \frac{2(N+1)}{N} \sum_{i=1}^{N} \frac{1}{i} - 3$$