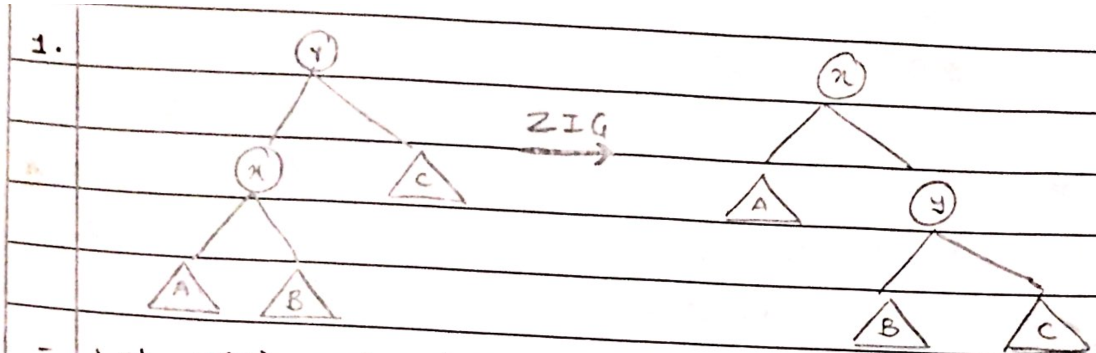
1.



- Let $r(x)$ and $r(y)$ be the ranks of x and y respectively, before ZIG operation.
- After ZIG operation → $r'(x)$ and $r'(y)$ be the new ranks of the above BST after splaying (x).
- The actual cost of rotation would be one time unit (rotation)

∴ For this case of Access Lemma would be;

~~Amortized cost = O~~

$$\text{Amortized cost} \leq 1 + r'(x) - r(x)$$

∴ Amortized cost for one rotation would turn out to be;

$$\rightarrow O\left(1 + \log \frac{s(y)}{s(x)}\right)$$

where $s(y)$ is the weight of the subtree rooted at 'y' and $s(x)$ is the weight of the subtree rooted at 'x'

Here, 'y' is the root node.

∴ For 'R' number of rotations,

(i) Amortized cost $= O\left(R + \frac{s(y)}{s(x)}\right)$ .... when x is present in the BST

(ii) Amortized cost $= O\left(R + \log \frac{s(y)}{\min\{s(x^-), s(x^+)\}}\right)$ ... when x is not present

where x^- → predecessor of x

x^+ → successor of x

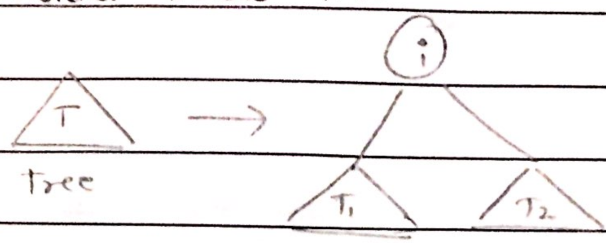
- All these operations have a logarithmic amortized time bound

- Consider 'W' as total weights of items in the tree and 'w' be the weight of child nodes, then for R rotations;

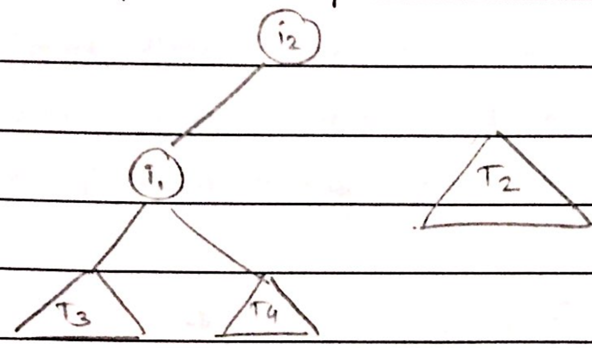
(i) Amortized cost $= O\left(R + \log \frac{W}{w(x)}\right)$ — if x is present

(ii) Amortized cost $= O\left(R + \log \frac{W}{\min\{w(x^-), w(x^+)\}}\right)$ — if x is not present

2. Consider below tree



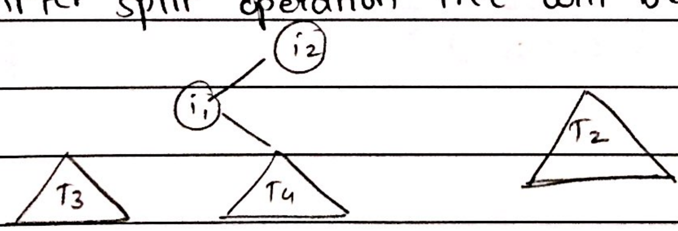
- split (i_1, T_1) → split at node i_1 in T_1



- This split results into All elements in T_3 < All elements in T_4

where node $\{i_1\}$ is root now.

∴ After split operation tree will be ;



- Since i_1 is the root element, elements in T_4 will be greater.

i.e. T_4 will have all nodes i, such ~~tat~~ that

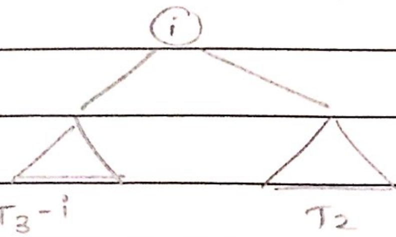
$$i \geq i_1$$

Hence the condition is satisfied $\boxed{i_1 \leq i \leq i_2}$

∴ Amortized cost $= O(1) + 3 \log \left(\frac{W(T_1)}{w(i_2)} \right)$

- Now we join (T_3, T_2) - move largest element from T_3 to root and join T_2 as a right child of root. P.T.O.

∴ After join (T_3, T_2) operation.



∴ Amortized cost $= O(1) + 3\log\left(\frac{W(T_3)}{w(i)}\right)$

∴ Total Amortized cost of cut (i_1, i_2, T) after split$\{i_1\}$, split$\{i_2\}$ and join$\{i\}$ is ;

$$= O(1) + 3\log\left(\frac{W(T)}{w(i_2)}\right) + O(1) + 3\log\left(\frac{W(T_1)}{w(i_1)}\right)$$

$$+ O(1) + 3\log\left(\frac{W(T_3)}{w(i)}\right)$$

$$\boxed{= O(1) + 3\log\left(\frac{W(T)}{w(i_2)}\right) + 3\log\left(\frac{W(T_1)}{w(i_1)}\right) + 3\log\left(\frac{W(T_3)}{w(i)}\right)}$$

3. (a) Let n be the height of T

- For complete binary tree with height of n, let

$S(n) = \sum_{\text{internal nodes } x \in T} 3^{-d(x)}$ ~~#~~ ~~4~~

$$\therefore \quad S(n) = \sum_{i=1}^{n} 2^{(i-1)} 3^{-i} \qquad \text{——— } (1)$$

So,

$$2S(n) = \sum_{i=1}^{n} 2^{i} \cdot 3^{-i} = 2\left(1 - \frac{2^{n}}{3}\right) \le 2 \qquad \text{——— } (2)$$

- From equation (2), we can get that $S(n) \le 1$ for all n.

$\therefore$ For arbitrary we can get ;

$$\sum_{\text{internal nodes } x \in T} 3^{-d(x)} \le 1$$

which is the pseudo - kraft inequality.

- And, we know that $\lim_{n \to \infty} S(n) = 1$

3. (b) To prove amortized cost of accessing item x

$= O(1 + d(x))$

where $d(x)$ - depth of node x

$w(x) = 3^{-d(x)} = 1/3^{d(x)}$

Given $d(root) = 1$

$\therefore$ Size of the root node $W = \sum_{\forall x \in T} \frac{1}{3^{d(x)}}$

- At depth n, BST has 2^{n-1} nodes.
- Similarly, a splay tree can have atmost 2^{i-1} nodes at depth i where i ranges from 1 to ∞

$\therefore$ Size of root node $(W) = \sum_{i=1}^{\infty} \frac{2^{i-1}}{3^i} = \frac{1}{3} \sum_{i=1}^{\infty} \frac{2^{i-1}}{3^{i-1}}$

P.T.O.

$\therefore \quad W = \frac{1}{3}\left(1 + \frac{2}{3} + \frac{2^2}{3^2} + \frac{2^3}{3^3} \dots\right) = \underline{\underline{1}}$

- Now, as we know Amortized cost $= O\left(1 + \log \frac{W}{w(x)}\right)$

On substitution, we get

Amortized cost $= O\left(1 + \log \frac{1}{3^{-d(x)}}\right)$

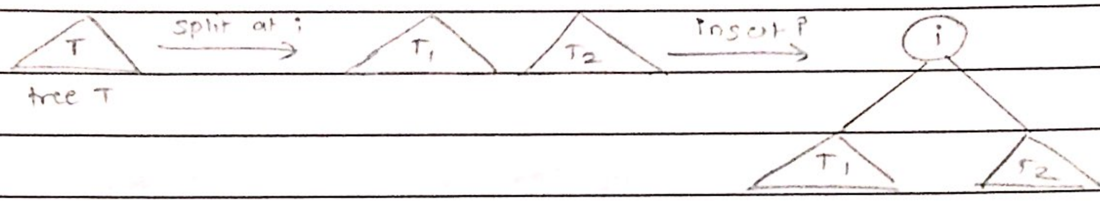
$= O\left(1 + \log 1 + d(x) \cdot \log 3\right)$

$= O\left(1 + d(x) \log 3\right)$

$= O\left(1 + d(x)\right) \quad \{\text{since } \log 3 = 0.477\}$

- Hence proved.

4. Consider tree T and node i that is to be inserted.



- split is required in order to insert i^{th} item / node
- split (T, i) and make a new tree whose left and right branches are trees T_1 and T_2 respectively
- A node i is added as the root / item to trees T_1 & T_2 (as depicted in the above diag.)
- The Amortized cost of insert operation with reference to the table is $= 3 \log \left(\frac{W - w(i)}{\min \{ w(i^-), w(i^+) \}} \right) + \log \left(\frac{W}{w(i)} \right)$

$$+ O(1) \qquad \text{—— } (1)$$

where W = weight of whole tree T
w = weight of child node / subtree rooted at i^{th}

- Derivation to derive equation (1);

• The normal split of tree at i operation takes

$$3 \log \left(\frac{W - w(i)}{\min \{ w(i^-), w(i^+) \}} \right) \qquad \text{—— } (2)$$

- $\omega(i^-)$, $\omega(i^+)$ are splayed to the root since initially i item was not present. Therefore total weight would be $[W - \omega(i)]$
- Linking nodes T_1 and T_2 requires $O(1)$ work —— (3)
- Linking T_1 and T_2 increases the potential, so potential of i also changes as it turns out to be the root node.

Total weight $= W(T_1) + W(T_2)$

$$\therefore \Delta\phi = \log W - \log \omega(i)$$

$$= \log \frac{W}{\omega(i)} \quad \text{—— (4)}$$

- Combining equations (2), (3) and (4) gives,

Amortized cost $= 3 \log \left(\frac{W - \omega(i)}{\min\{\omega(i^-), \omega(i^+)\}} \right) + \log \left(\frac{W}{\omega(i)} \right) + O(1)$