ASSIGNMENT: HW 4

NAME: PRASANTH BHAGAVATULA

FALL 2015

CWID: A20355611

(1) In this simple-minded splaying, we just use one-rotation at a time and so the amortized cost of one rotation is

$$\hat{c} = 1 + r'(x) - r(x)$$

where $r'(x)$ is the rank after the rotation and $r(x)$ is the rank before the rotation. Hence, for an access, the total amortized cost is

$$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} \left(1 + \phi_i - \phi_{i-1}\right)$$

$$\Rightarrow \sum_{i=1}^{n} \hat{c}_i = n + \phi_n - \phi_{n-1} + \phi_{n-1} - \phi_{n-2} \cdots + \phi_2 - \phi_1 + \phi_1 - \phi_0$$

$$\Rightarrow \sum_{i=1}^{n} \hat{c}_i = n + \phi_n - \phi_0$$

$$\Rightarrow \sum_{i=1}^{n} \hat{c}_i = n + r'(t) - r(x) \quad (\because x \text{ becomes the root}$$

$$\Rightarrow \sum_{i=1}^{n} \hat{c}_i = n + \log \frac{s(t)}{s(x)}$$

$$\Rightarrow \text{Amortized cost} = O\left(1 + \log \frac{s(t)}{s(x)}\right)$$

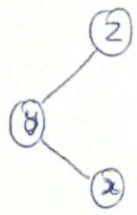
ASSIGNMENT: HW4

FALL 2015

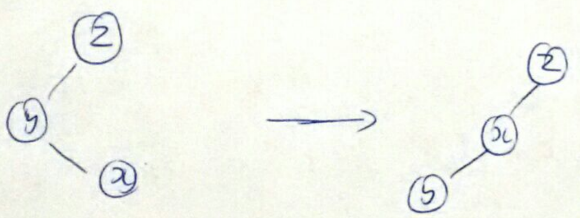
NAME: PRASANTH BHAGAVATULA

CWID: A20355611

2. We will apply the ZIG-ZAG operation: when x is the right child of a left parent



Going by the algorithm given for simple splaying, here $x = \text{right}(P(x))$ and $P(x) \neq \text{right}(g(x))$ so only "rotate left ($P(x)$)" will happen which means only a 'ZAG' operation is performed.



Now, x is left ($P(x)$) and only in case of $P(x) = \text{left}(g(x))$ we can get two rotations (rotate right ($g(x)$) and rotate right ($P(x)$)) and both are ZIG rotations and so, it is required for analysing this algorithm. Hence, after a ZIG-ZAG operation; we need to lookout for a "ZIG' operation (or) 'ZIG-ZIG' operation

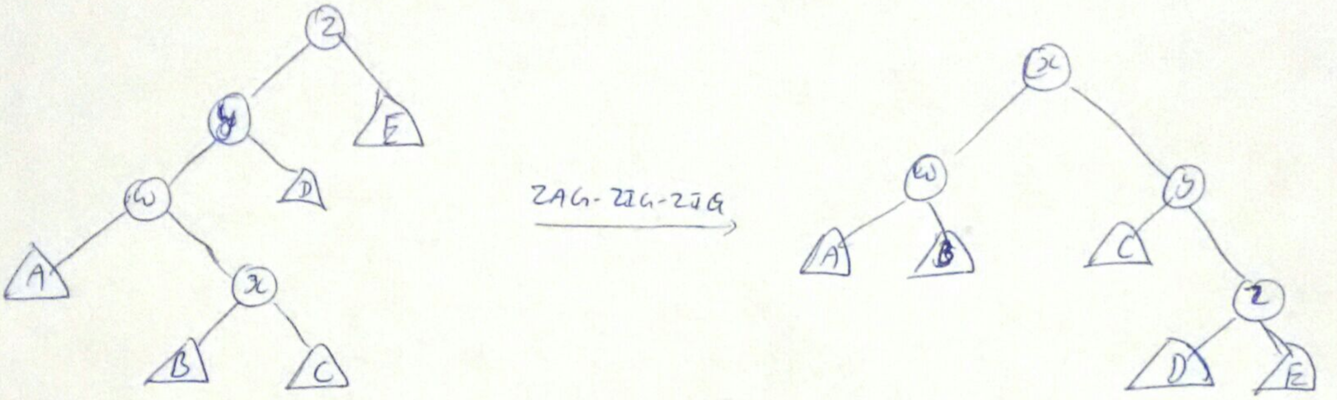
ASSIGNMENT: HW4

NAME: PRAVANTH BHAGAVATULA

FALL 2015

CWID: A20355611

③



The amortized cost of this operation is

$$\hat{c} = c + \Delta\phi ;$$

where c is the actual cost and $\Delta\phi$ is change in potential. Looking ath the figure; we can say that we have performed '3' rotations and so the actual cost is 3. Now the change in potential is new potential minus the old potential. and let $r(t)$ represents old potential of node 't' and $r'(t)$ represents new potential of node 't'. Since, there are no changes in sub-trees A, B, C, D, E; there is no change in potential for those sub-trees. Hence

$$\hat{c} = 3 + r'(x) + r'(y) + r'(w) + r'(z) - r(x) - r(y) - r(w) - r(z)$$

ASSIGNMENT: HW4

NAME: PRASANTH BHAGAVATULA

CWID: A 20355611

Before the operation, if we look at the tree; we can clearly say that $r(x) \leq r(w) \leq r(y) \leq r(z)$. Hence, we can re-write the equation as

$$\hat{c} = 3 + r'(x) + r'(y) + r'(w) + r'(z) - r(x) - r(y) - r(z) - r(w)$$

$$\leq 3 + r'(x) + r'(y) + r'(w) + r'(z) - r(x) - r(x) - r(x) - r(x)$$

$$\leq 3 + r'(x) + r'(y) + r'(w) + r'(z) - 4r(x)$$

After the operation, if we look at the new tree, we can clearly say that $r'(y) \geq r'(z)$. So, we re-write the equation as

$$\hat{c} \leq 3 + r'(x) + r'(y) + r'(w) + r'(z) - 4r(x)$$

$$\leq 3 + r'(x) + r'(y) + r'(w) + r'(y) - 4r(x)$$

$$\leq 3 + r'(x) + 2r'(y) + r'(w) - 4r(x)$$

Hence;

$$\hat{c} \leq 3 + r'(x) + 2r'(y) + r'(w) - 4r(x)$$

ASSIGNMENT: HW 4

NAME: PRASANTH BHAGAVATULA

FALL 2015

CWID: A20355611

④ After the operation, if we look at the new tree structure, we can write $s'(x)$ as

$$s'(x) = w(x) + s'(w) + s'(y)$$

that is, sum of weight of node 'x' and weight of subtrees at node 'w' and 'y'. Hence $s'(x) > s'(w) + s'(y)$.

Consider that $\dfrac{s'(w)}{s'(x)} = \dfrac{1}{3}$.

Then based on the above inequality, we can say that

$$\frac{s'(y)}{s'(x)} \leq \frac{2}{3}.$$

Now consider the equation 1, given in problem

$$2\,r'(y) + r'(w) - 3\,r'(x) = \log \frac{s'(y)}{s'(x)} + \log \frac{s'(w)\,s(y)}{s(x)^2}$$

$$= \log\left(\frac{s'(y)}{s(x)}\right) + \log\left(\frac{s'(w)}{s'(x)} \times \frac{s(y)}{s(x)}\right)$$

$$\leq \log\left(\frac{2}{3}\right) + \log\left(\frac{1}{3} \times \frac{2}{3}\right)$$

$$\leq \log \frac{2}{3} + \log \frac{2}{9}$$

ASSIGNMENT: HW4

NAME: PRASANTH BHAGAVATULA

FALL 2015

CWID: A20355611

(6) The delete operation can also be done by searching for the node containing 'i'. Let this node be x and the parent of x be 'y'. We then replace x as a child of y by joining left and right subtrees of x and then by splaying on y.

So this operation involves accessing node containing 'i' i.e. x and ~~splaying~~ on joining subtrees ~~and~~ of 'x'.

For accessing element ~~in~~ 'i' the access (i, T) will be

$$access(i,T) = 3 \log\left(\frac{W}{w(i)}\right) + 1$$

and for joining sub trees T_1 and T_2, so

$$join(T_1, T_2) = 3 \log\left(\frac{W - w(i)}{w(i^-)}\right) + O(1)$$

since node is ~~also~~ deleted while joining the sub-trees

Hence the total time

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

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

ASSIGNMENT : HW4

NAME: PRASANTH BHAGAVATULA

CWID : A20355611

FALL2015

(5) Using the equation in problems 3 and 4,

$$\hat{c} \leq 3 + n'(x) + 2n'(y) + n'(w) - 4n(x)$$

and from problem 4.

$$2n'(y) + n'(w) - 3n'(x) \leq \alpha$$

$$\Rightarrow 2n'(y) + n'(w) \leq \alpha + 3n'(x)$$

applying the above inequality in the first equation

$$\hat{c} \leq 3 + n'(x) + \alpha + 3n'(x) - 4n(x)$$

$$\Rightarrow \hat{c} \leq 3 + \alpha + 4(n'(x) - n(x))$$

$$\Rightarrow \hat{c} \leq (3+\alpha) + (1+3\times 1)(n'(x) - n(x))$$

If we look at the multiplying factor of $(n'(x) - n(x))$, we can multiply 3 using 'β' (>1) and so, we can say that

$$\hat{c} \leq (1+3\beta)(n'(x) - n(x))$$