Splay trees [Sleator - Tarjan]

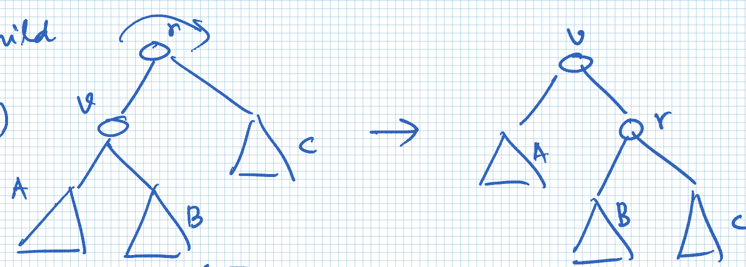
- Binary Search Tree (left subtree ≤ node ≤ right subtree) [INSERT/DELETE/FIND]
- Height Balancing NOT explicit but performed through "ying" operations
- Comparison to other height balanced trees
  - $O(\log n)$ amortized (as against worst-case) running time per operation
  - simpler - only operation "splaying"
  - has additional properties (see PSET 1)

SPLAY operations (splay(v))

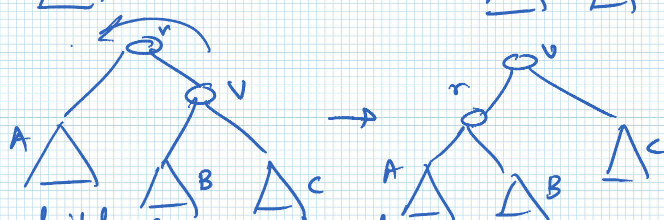
Case 1 - v is the child of the root node

reversible

- left child (right rotate)

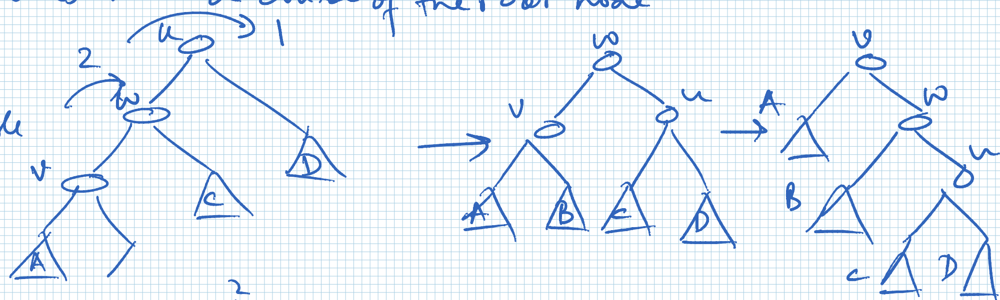


- right child (left rotate)

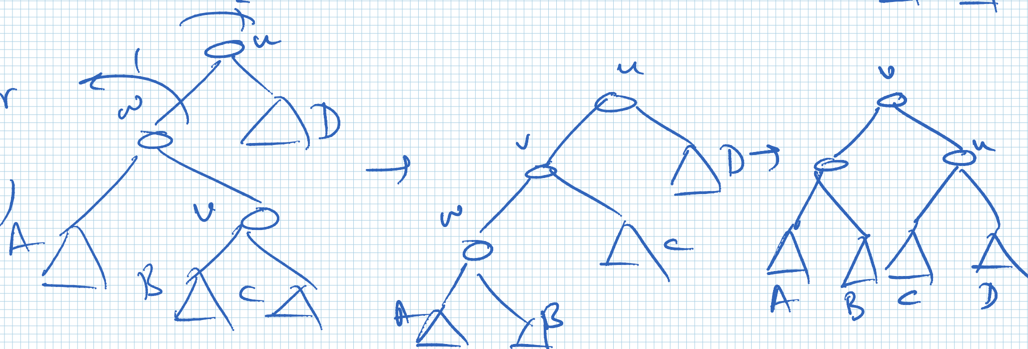


Case 2 : v is not a child of the root node

ll (reversible with rr)



lr (not reversible)



Analysis

Let x be a node

s_x = # of descendants of x including x itself

rank $r_x = \log s_x$

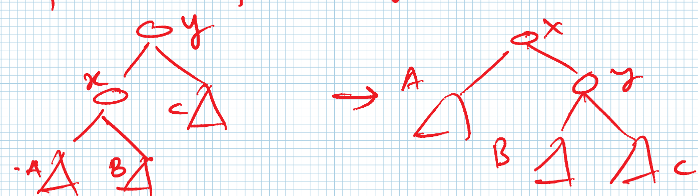
Potential $\Phi_T = \sum_{x \in T} r_x$

Lemma: Amortized cost of single level splay for node x

$\leq (r'(x) - r(x)) + 1$ where r' and r are ranks of x

after and before splay respectively.

Proof:



Real cost = 1

$\Delta\phi = r'(x) + r'(y) - r(x) - r(y)$

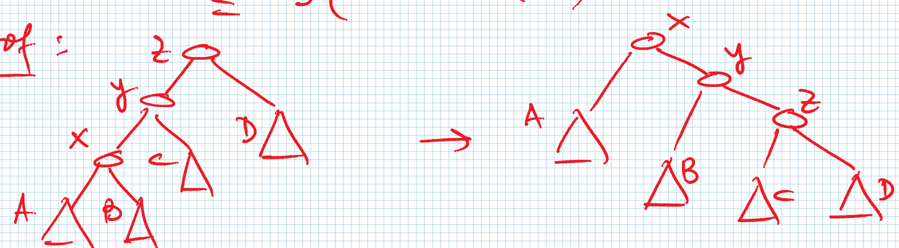
$= (r'(x) - r(x)) + (r'(y) - r(y))$

$\leq r'(x) - r(x)$ since: $r'(y) \leq r(y)$

Lemma: Amortized cost of an ll (or rr) splay at x

$\leq 3(r'(x) - r(x))$

Proof:



Actual cost = 2

$\Delta\phi = (r'(x) + r'(y) + r'(z)) - (r(x) + r(y) + r(z))$

$= (r'(y) + r'(z)) - (r(x) + r(y))$ [since $r'(x) = r(z)$]

$\leq (r'(x) + r'(z)) - (r(x) + r(x))$ [since $r'(x) \geq r'(y)$, $r(x) \leq r(y)$]

Using concavity of log function,

$$\frac{r(x) + r'(z)}{2} = \frac{\log s_x + \log s'_z}{2} \leq \log\left(\frac{s_x + s'_z}{2}\right) \leq \log\frac{s'_x}{2}$$

$= r'(x) - 1$

$\Rightarrow r'(z) \leq 2r'(x) - 2 - r(x)$

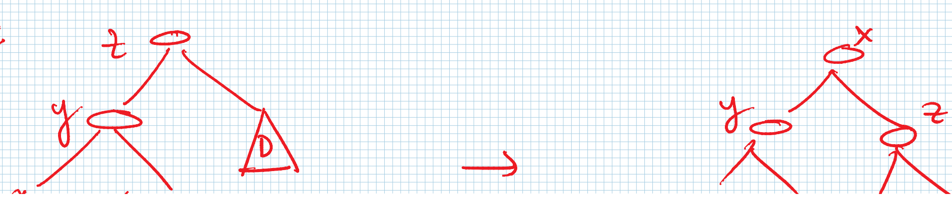
$\Delta\phi \leq 3r'(x) - 2 - r(x) - 2r(x) = 3(r'(x) - r(x)) - 2$

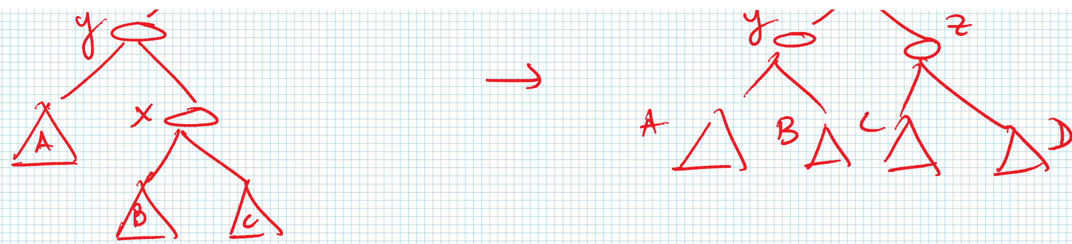
Overall, actual cost $+ \Delta\phi \leq 3(r'(x) - r(x))$

Lemma: Amortized cost of an lr (or rl) splay at vertex x

$\leq 2(r'(x) - r(x))$

Proof:





Actual cost $= 2$

$$\Delta\phi = (r'(x) + r'(y) + r'(z)) - (r(x) + r(y) + r(z))$$

$$= (r'(y) + r'(z)) - (r(x) + r(y)) \quad [r'(x) = r(z)]$$

$$\le (r'(y) + r'(z)) - (r(x) + r(x)) \quad [r(y) \geq r(x)]$$

By concavity of log function,

$$\frac{r'(y) + r'(z)}{2} \le \log\left(\frac{s'(y) + s'(z)}{2}\right) \le \log\left(\frac{s'(x)}{2}\right)$$

$$= r'(x) - 1$$

$$\Rightarrow \quad r'(y) + r'(z) \le 2r'(x) - 2$$

Thus, $\Delta\phi \le 2r'(x) - 2 - 2r(x)$

Overall, actual cost $+ \Delta\phi \le 2(r'(x) - r(x))$

Operations

- Insert : insert in BST and splay up to root
- Delete : find pred/succ and splay up to root
  Element becomes leaf
  Delete element
- Find : find element and splay up to root

Theorem : The amortized cost of each operation in a splay tree is $O(\log n)$.

Proof : amortized cost $\le \sum 3(r^{i+1}(x) - r^{i}(x)) + 1$

$$= 3(r^{final}(x) - r^{init}(x)) + 1$$

$$\le O(\log n)$$