Advanced Data Structure

Splay Trees (part 2)

Recap

- HW - due Friday

Analysis via Potential Method.

Idea: Let -

- D_i = data structure after i^{th} query
- C_i = cost of i^{th} operation
  so varies! splay: depended on tree
- and $\Phi(D)$ is potential of a data structure
  (used size/rank)

Define A_i as follows:

$$A_i = C_i + \Phi(D_i) - \Phi(D_{i-1})$$

Note: - Φ is arbitrary
- A_i is wierd & not actual cost

(bear with me...)

Using this werd Φ + A_i :

$$\sum_{i=1}^{m} c_i = \text{cost of all } m \text{ operations}$$

$$= \sum_{i=1}^{m} \Big(c_i + \Phi(D_i) - \Phi(D_i) + \Phi(D_{i-1}) - \Phi(D_{i-1}) \Big)$$

Why? Regroup:

$$= \sum_{i=1}^{m} \overbrace{\Big(c_i + \Phi(D_i) - \Phi(D_{i-1}) \Big)}^{A_i} - \sum_{i=1}^{m} \Big(\Phi(D_i) - \Phi(D_{i-1}) \Big)$$

Then use: $A_i = c_i - \Phi(D_{i-1}) + \Phi(D_i)$

$$= \sum_{i=1}^{m} A_i + \Phi(D_0) - \Phi(D_m)$$

Take away!

$$\sum_{i=1}^{m} C_i = \sum_{i=1}^{m} A_i + \Phi(D_0) - \Phi(D_m)$$

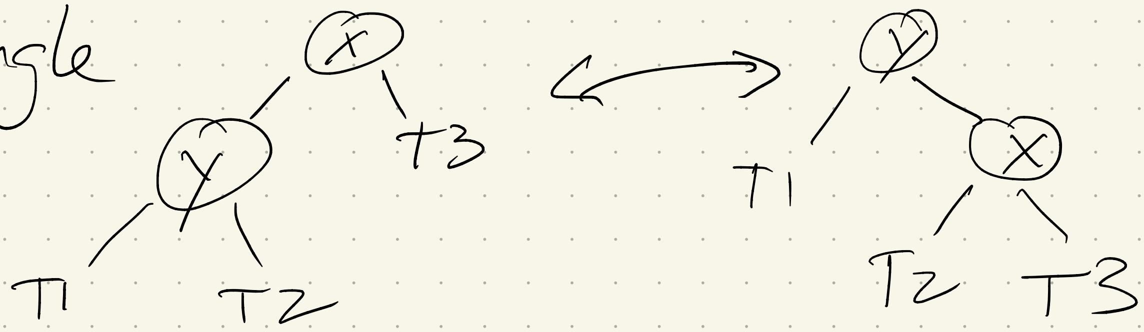
$\Phi(D_0) - \Phi(D_m)$ is called "net drop in potential"

<u>So</u>: if we can provide a bound on A_i (or $\sum A_i$), that + potential drop is cost of our m ops.

Last time: splaying

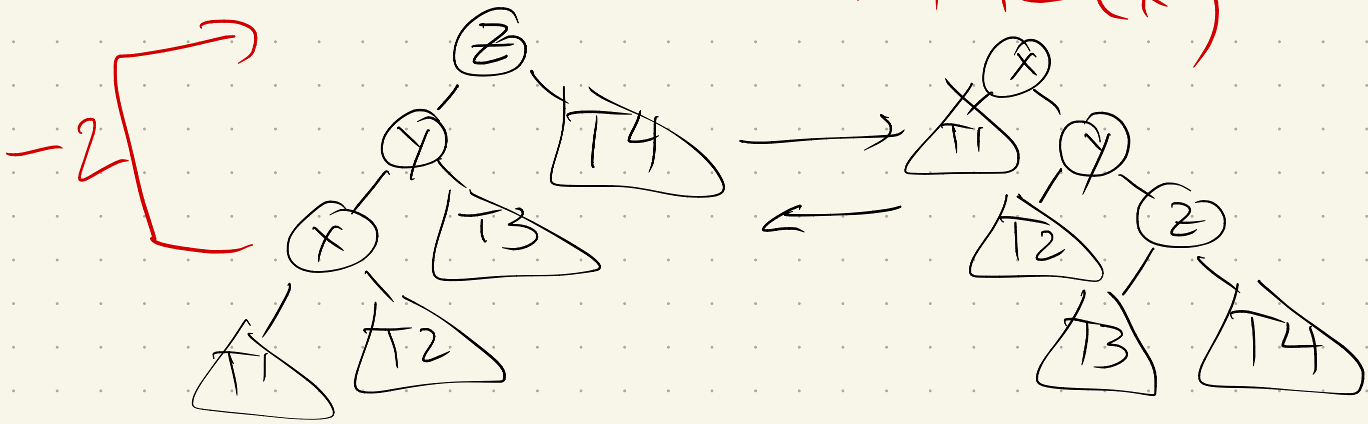
Rotations:

- single



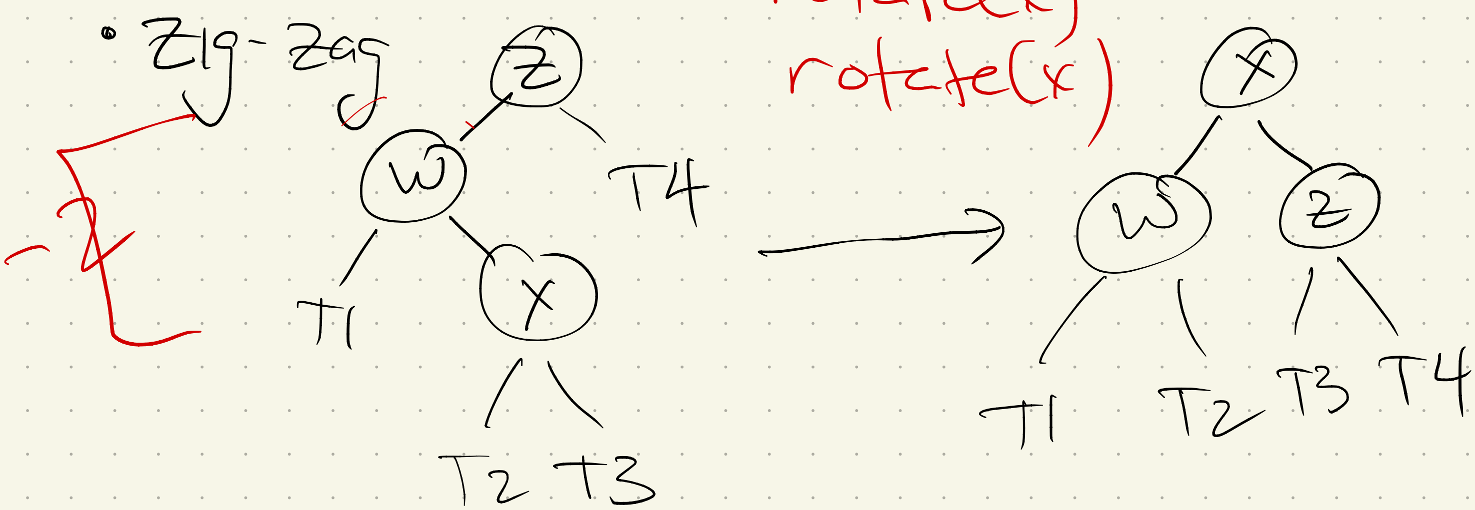
- roller coaster:

rotate(y)
rotate(x)



- zig-zag

rotate(x)
rotate(x)



Code :

Splay (x) : x is the root

While (x ≠ root) or (parent(x) ≠ root)

  double rotation (x) ← either zigzag or rollercoaster

If x ≠ root

  rotate(x)

Search (x) :

node ← BSTFind (x)

  (assume this returns x, or pred/succ if x is not in tree)

splay (node)

Insert(x)

node ← BSTInsert(x)

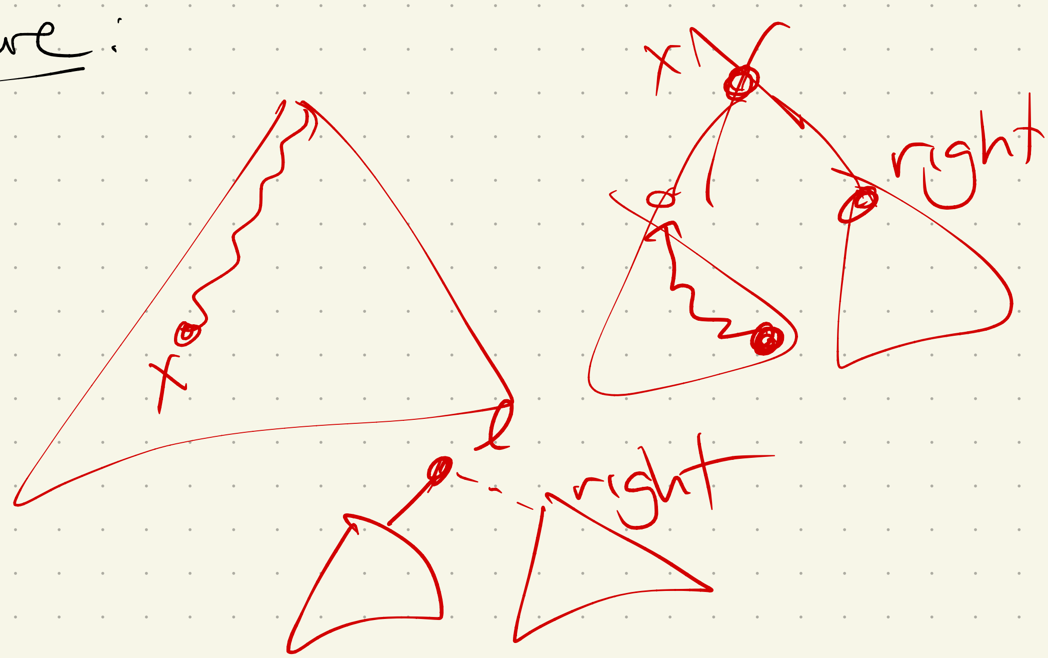
  (assume this returns x's node in tree)

splay(node)

Code (cont)

```
Delete(x):
    xnode ← BSTFind(x)
    if xnode.value = x:
        splay(xnode)
        left ← (xnode.left)
        right ← (xnode.right)
        delete(xnode)
        l ← FindLargest(left)
        splay(l)
        l.right ← right
```

Picture:



To get our Φ:

T = Binary tree w/ n nodes, labeled $1..n$

Each node v has weight $w(v)$

- $w(v) > 0$ (today: $w(v) = 1$)

Let $s(v) = w(v) + s(v.\text{left}) + s(v.\text{right})$

$r(v) = \lg(s(v))$

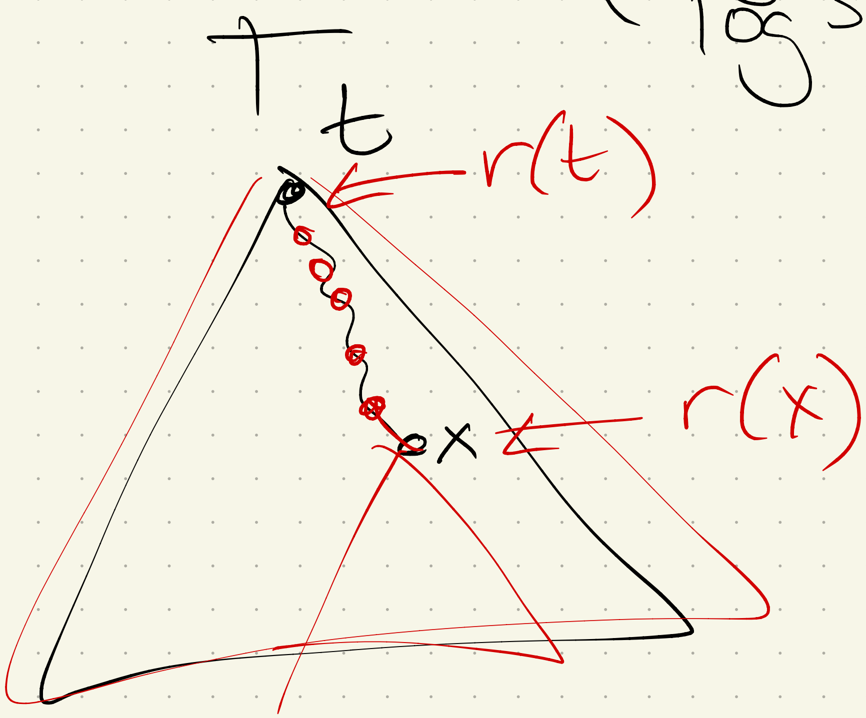
$$\phi(T) = \sum_{i=1}^{n} \log(s(i))$$

$$= \sum_{i=1}^{n} r(v)$$

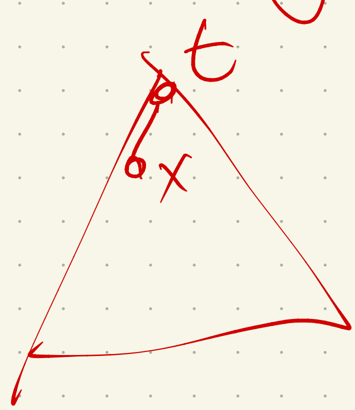
Access Lemma:

Amortized time to splay a tree T w/ root t at a node x is $\leq 3(r(t) - r(x)) + 1$

$$= O\left(\log \frac{s(t)}{s(x)}\right)$$



$\log a - \log b = \log \frac{a}{b}$



pf: If $x = t$, trivially true (and $= 1$)

If not, need to splay x up to t

Consider one rotate in splay:

Let $\left.\begin{matrix} r_0(i), r_1(i) \\ s_0(i), s_1(i) \end{matrix}\right\} = \left\{\begin{matrix} \text{rank} \\ \text{size} \end{matrix}\right\}$ of node i

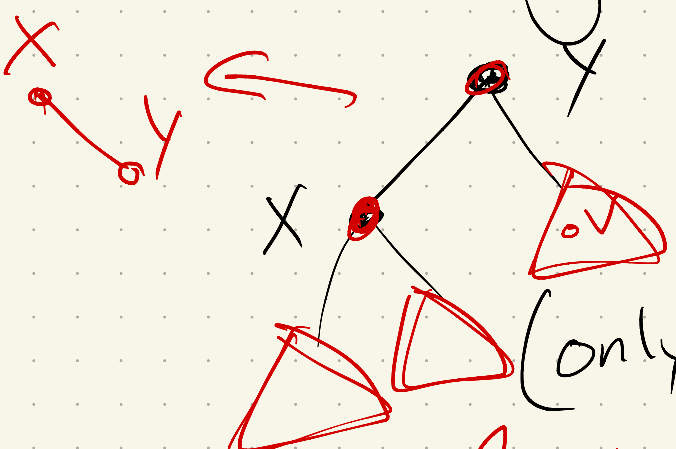
$\underbrace{}_{\text{before}}$ / $\underbrace{}_{\text{after}}$ a rotate

recall $a = t + \overset{+V}{\Phi_1(T)} - \underline{\overset{-V}{\Phi_0(T)}}$

where t = cost of op

Φ = potential (before + after)

- If single rotation: we're at the root!



$a = \underline{\Phi_1(T) - \Phi(T)}$

(only x + y change!)

$= 1 + r_1(x) + r_1(y) - r_0(x) - r_0(y)$

(by definition of Φ)

$1 + r_1(x) + \underline{r_1(y)} - r_0(x) - \underline{r_0(y)}$

Now: y rotates <u>down</u>

so $\underline{r_1(y)} \le \underline{r_0(y)}$

$\Rightarrow \underbrace{r_1(y) - r_0(y)} \le 0$

$\Rightarrow \le 1 + \underbrace{r_1(x) - r_0(x)} - 0$

& after rotation, x is root, so $r_1(x) \ge r_0(x)$

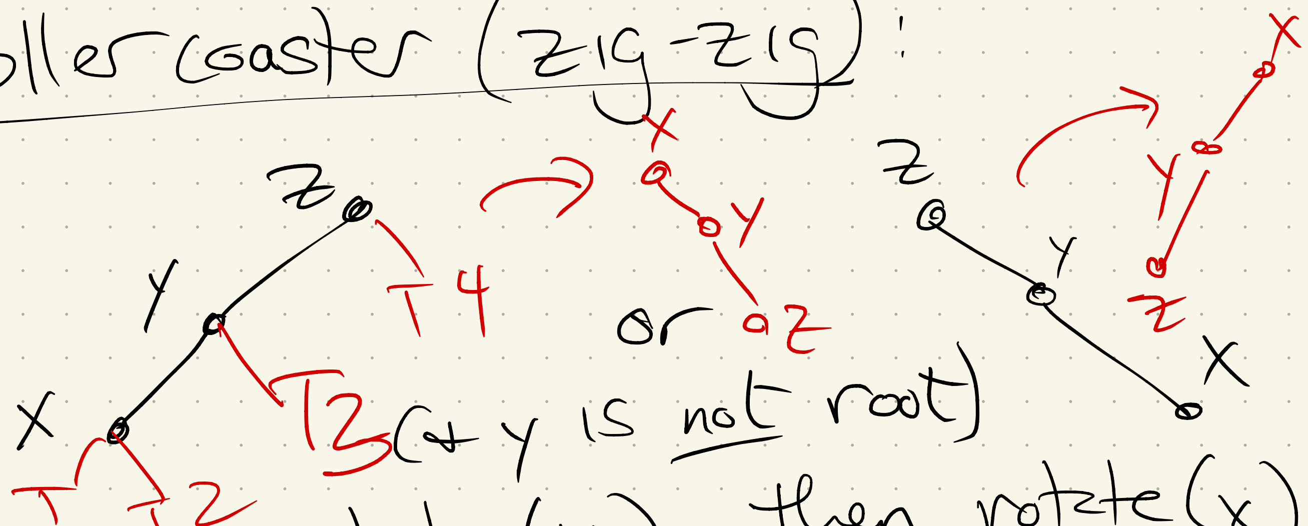
$\Rightarrow r_1(x) - r_0(x) \ge 0$

$\Rightarrow \le 1 + 3\left(r_1(x) - r_0(x)\right)$

(adding extra for fun)

• If double rotation, 2 cases

1) Rollercoaster (zig-zig):



or

(& y is not root)

We rotate(y), then rotate(x)

actual cost: 2 $\quad t+\Phi_1-\Phi_0$

amortized cost =

$$2+\cancel{r_1(x)}+r_1(y)+r_1(z)-r_0(x)-r_0(y)-\cancel{r_0(z)}$$

• x moves up

• y & z move down

More carefully: $r_1(x)=r_0(z)$

$$=2+r_1(y)+r_1(z)-r_0(x)-r_0(y)$$

& $r_1(x)\ge r_1(y)$ & $r_0(x)\le r_0(y)$

$$\Rightarrow\ \le 2+r_1(x)+r_1(z)-2r_0(x)$$

Ok (still R.C. rotate)

Have: $2 + r_1(x) + r_1(z) - 2r_0(x)$

& goal is $\le 3(r_1(x) - r_0(x))$

Manipulate:

True if

$$r_1(z) + r_0(x) - 2r_1(x) \le -2$$

finish Wed!

Now: $r_1(z) + r_0(x) - 2r_1(x)$

apply dfn: $r_i(v) = \log(s_i(v))$

& $\log b + \log c = \log(bc)$

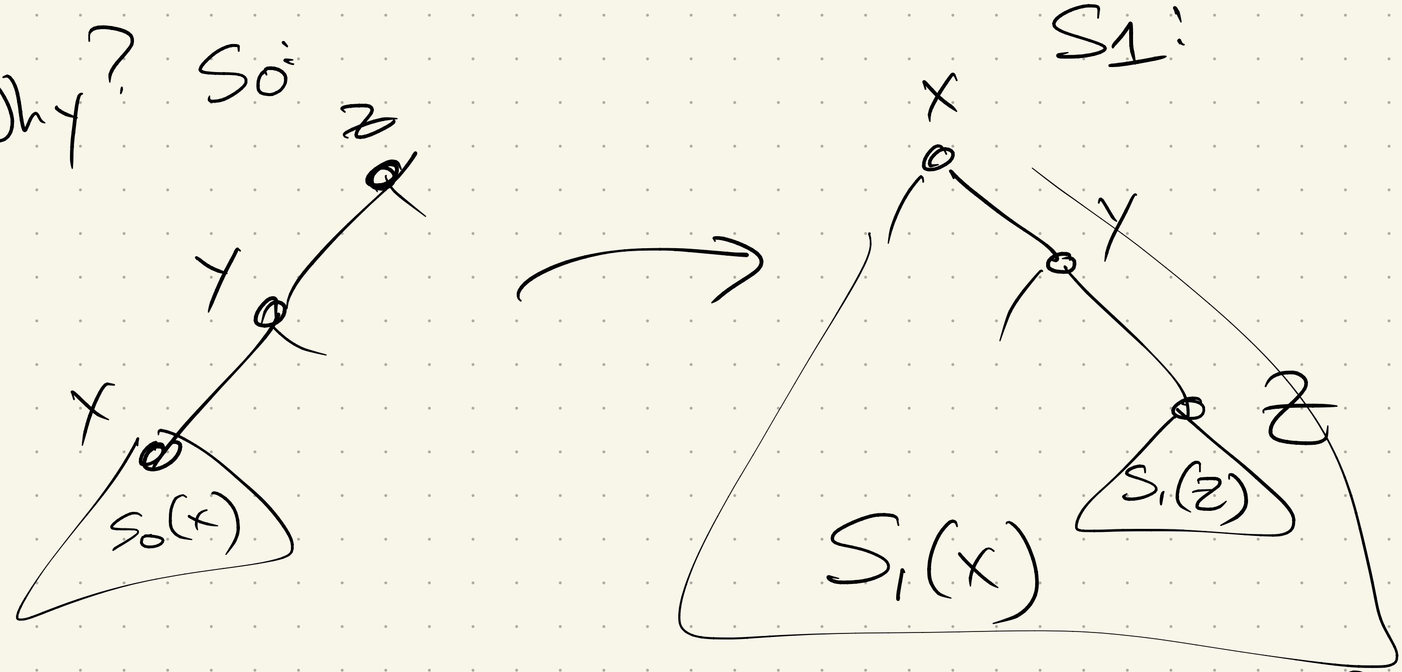
$\Longrightarrow$

<u>Now</u>: $\log\left(\underbrace{\frac{S_1(z)}{S_1(x)}}_{b} \cdot \underbrace{\frac{S_0(x)}{S_1(x)}}_{c}\right)$

let $b = $ ⤴ $\quad c = $ ⤴

Observe: $b + c \leq 1$

Why?



so $S_1(x) \geq S_0(x) + S_1(z)$

How big can this $\log(b \cdot c)$ get?

Calculus $\Rightarrow$ max when $b = c = \frac{1}{2}$

so $\leq \log\left(\frac{1}{4}\right)$

Result: Each roller coaster has amortized cost

$$\leq 3(r_1(x) - r_0(x))$$

2) Zig-zag: same math tricks (Notes are posted)

Result: Suppose k rotates to splay:

$$3r_k(x) - 3r_{k-1}(x) + 1$$

$$+ \sum_{j=1}^{k-1} 3(r_j(x) - r_{j-1}(x))$$

$$=$$

Result: Balance Thm:

Given a sequence of m accesses to an n-node splay tree, total run time is

$$O(n \log n + m \log n).$$

Proof: Assign $w(v) = 1$ to every node v.

Note: $1 \le S(x) \le n$

Splay cost:

$$\le 3(r(t) - r(x)) + 1$$

Max potential Φ =

$$\sum_{i=1}^{n} \log(s(i)) \le \sum_{i=1}^{n}$$

Min potential :

$$\sum_{i=1}^{n} \log(s(i)) \ge$$

So $\Phi_{max} - \Phi_{min} =$

Next time:

Optimality Thm ...