Optimal BST

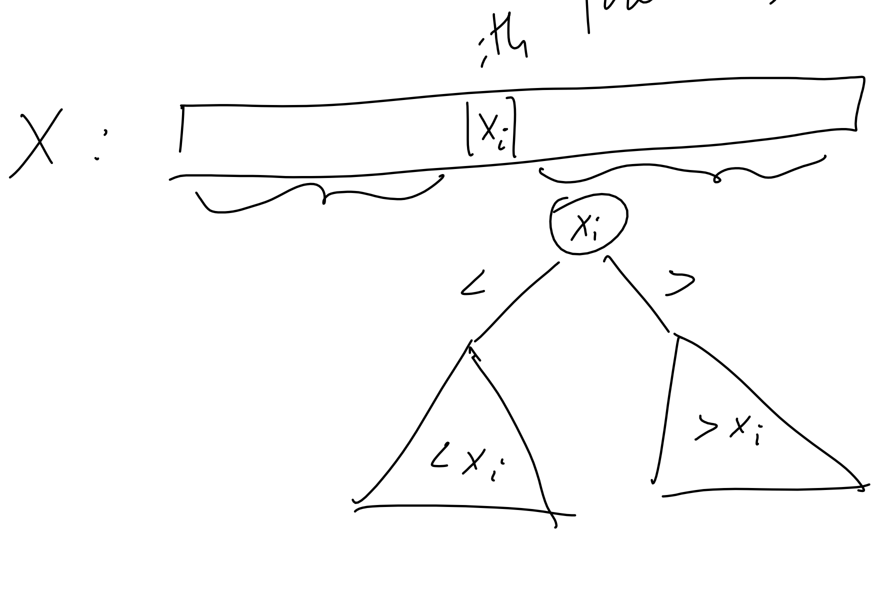
Given: set of n items

& Query set of size m

f_i : # of times i^{th} item is searched for

$$\sum_{i=1}^{n} f_i = m$$

Question: Construct a BST s.t. time to search for all queries is minimized

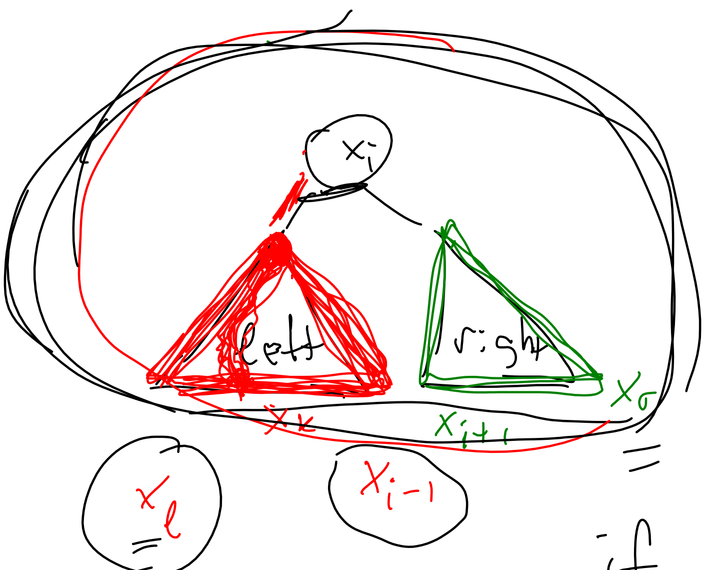


int optBST $(X[l,r])$ // the cost of searching for keys in the range $X[l,r]$ in the optimal tree on the items in this range

```
if l = r return f_l
if l > r return 0
min = +∞
for i = l to r
    // try x_i as the root
    left = optBST(X[l, i-1])
    right = optBST(X[i+1, r])
```

$$\text{cost} = \text{left} + \text{right} + 1 \cdot f_i + \sum_{k=l}^{i-1} f_k + \sum_{k=i+1}^{r} f_k$$

$$= \left(\sum_{k=l}^{r} f_k\right) + \text{left} + \text{right}$$




```
    if cost < min
        min = cost
return min
```

$$\text{optBST}(l, r) = \min_{i=l}^{r} \left(\sum_{k=l}^{r} f_k + \text{optBST}(l, i-1) + \text{optBST}(i+1, r) \right)$$

$$\left(\sum + \text{opt}(l,l) + \text{opt}(l+1,r)\right), \left(\sum + \text{opt}(l,l+1) + \text{opt}(l+2,r)\right) \ldots$$

$$\Rightarrow O(n^3)$$

Q — m queries

f_i — # of times x_i is searched for

$\left(\sum_{i=1}^{n} f_i = m\right)$

n — # of distinct items searched for

$$\boxed{p_i = \frac{f_i}{m}} \Rightarrow \sum_{i=1}^{n} p_i = 1$$

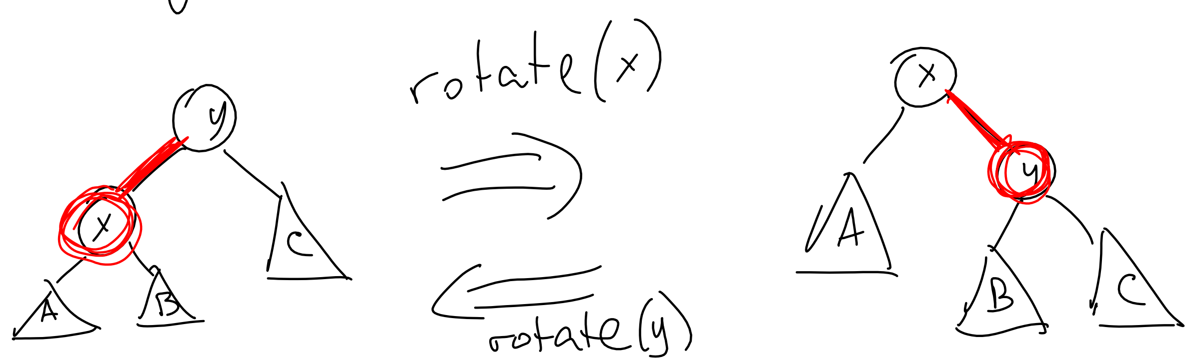
To perform queries of Q it takes

$$\Theta\left(m \underbrace{\sum_{i=1}^{n} p_i \cdot \log \frac{1}{p_i}}_{H_n}\right)$$

$$= \Theta(m \cdot H_n)$$

Splay trees

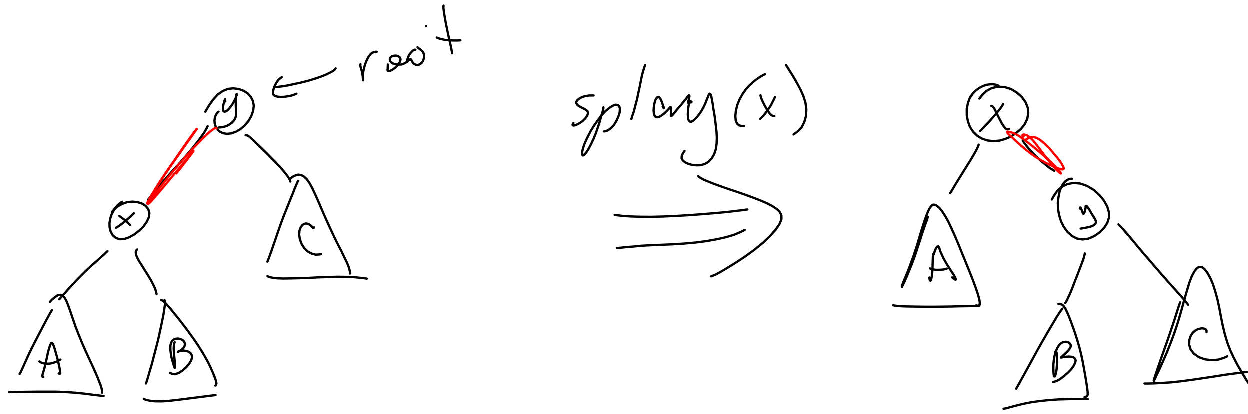
self-adjusting Binary Search trees



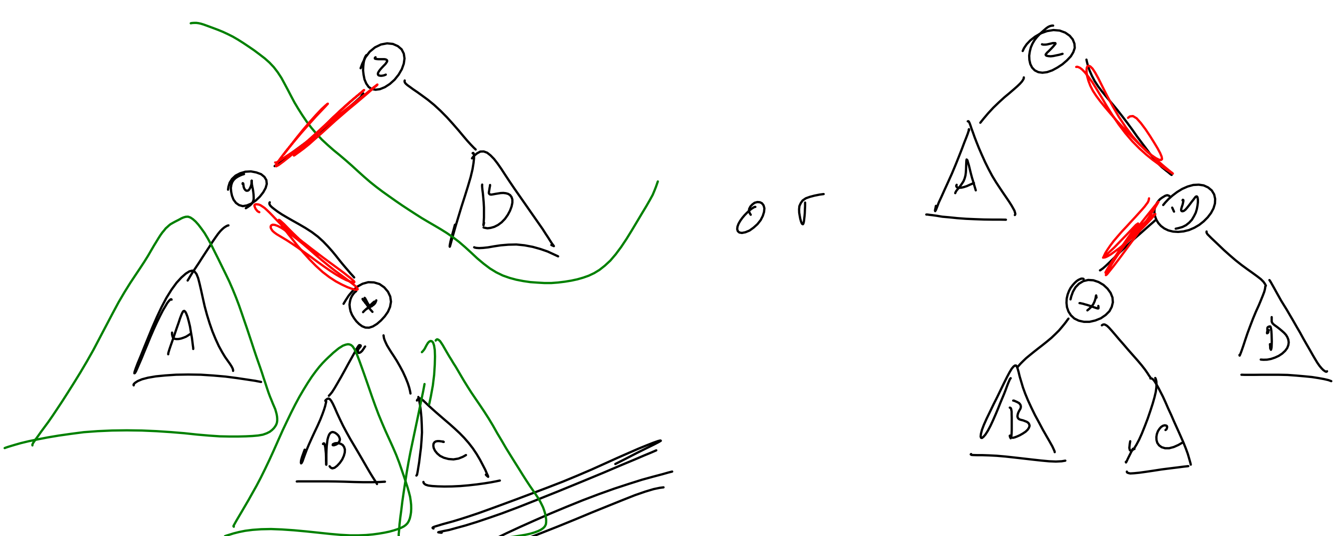
Splay trees: simple BST with add'l constraint that Any access to node v, brings v to the root via splay operations.

Splay(v) operation

1. zig: if x's parent is root: rotate(x)

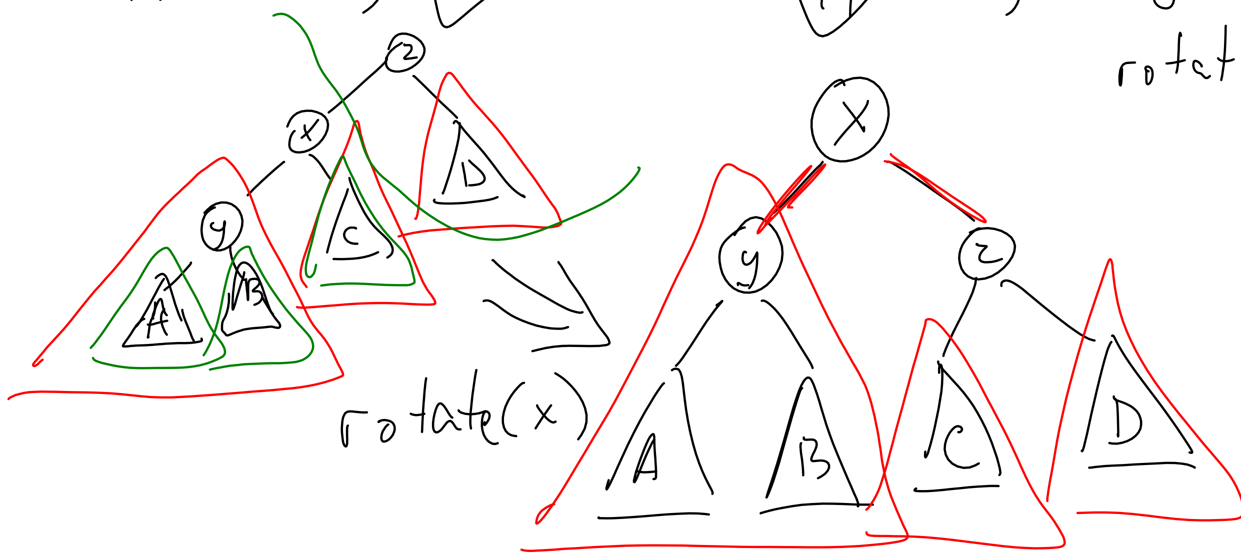


2. zig-zag:

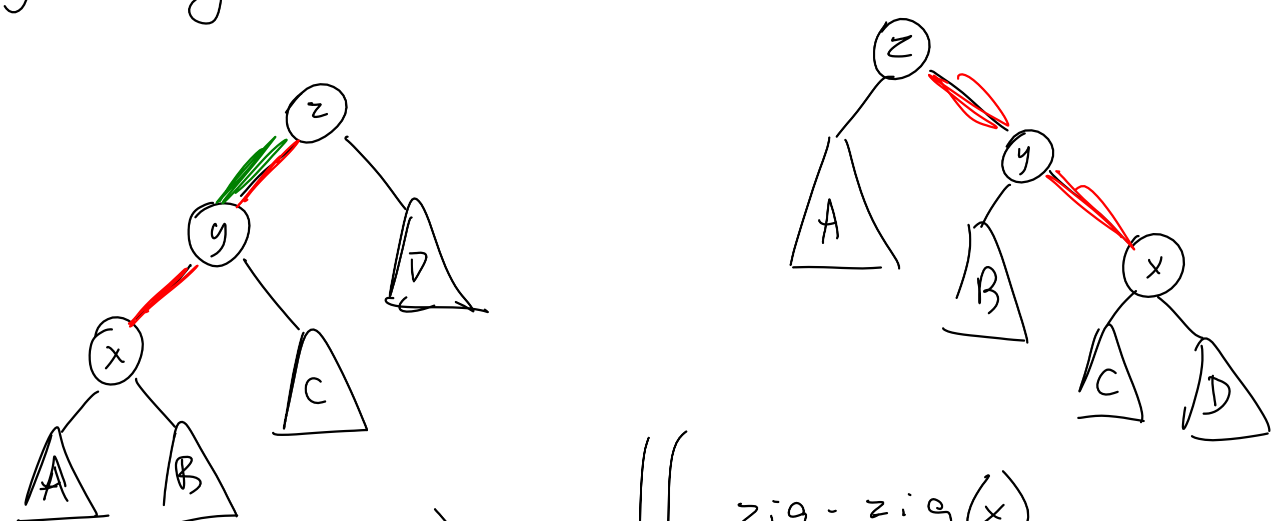


rotate(x)

zig-zag(x) :=
rotate(x); rotate(x)

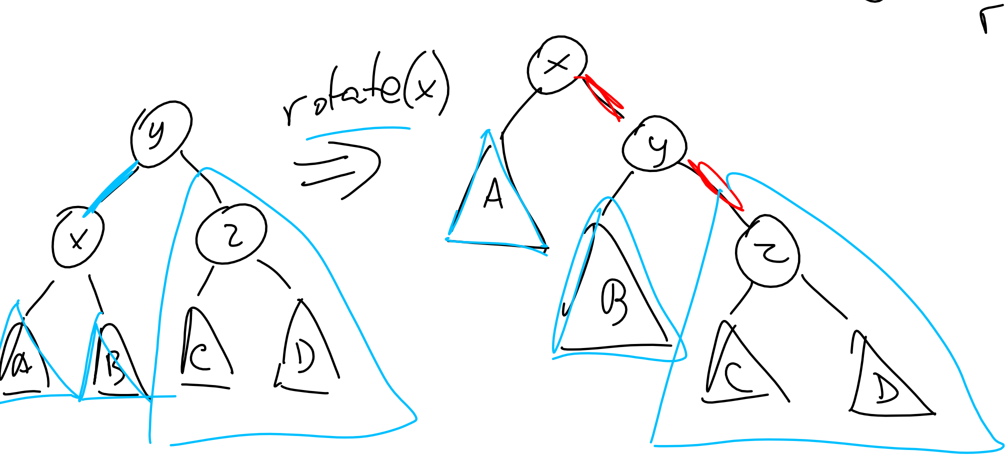


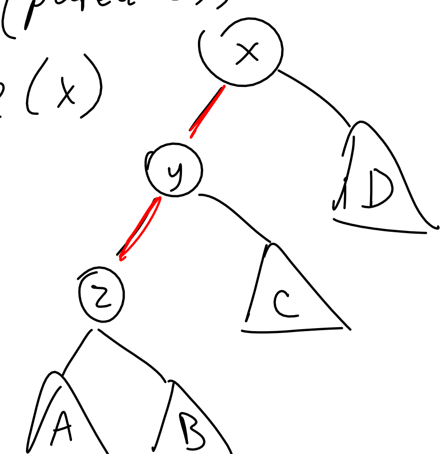
3. zig-zig



rotate(parent(x))

zig-zig(x)
rotate(parent(x))
rotate(x)





splay(x) $\Rightarrow$ moves x above its parent & grandparent

```
Search(key)
    x = BST-search(key)
    while (x ≠ root)
        splay(x)
    return x
```

Amortized Analysis of Searching on Splay trees.

$w(x)$ = weight of node x $(w(x) = 1)$

$s(x) = \sum_{v \in T(x)} w(v)$ ← size

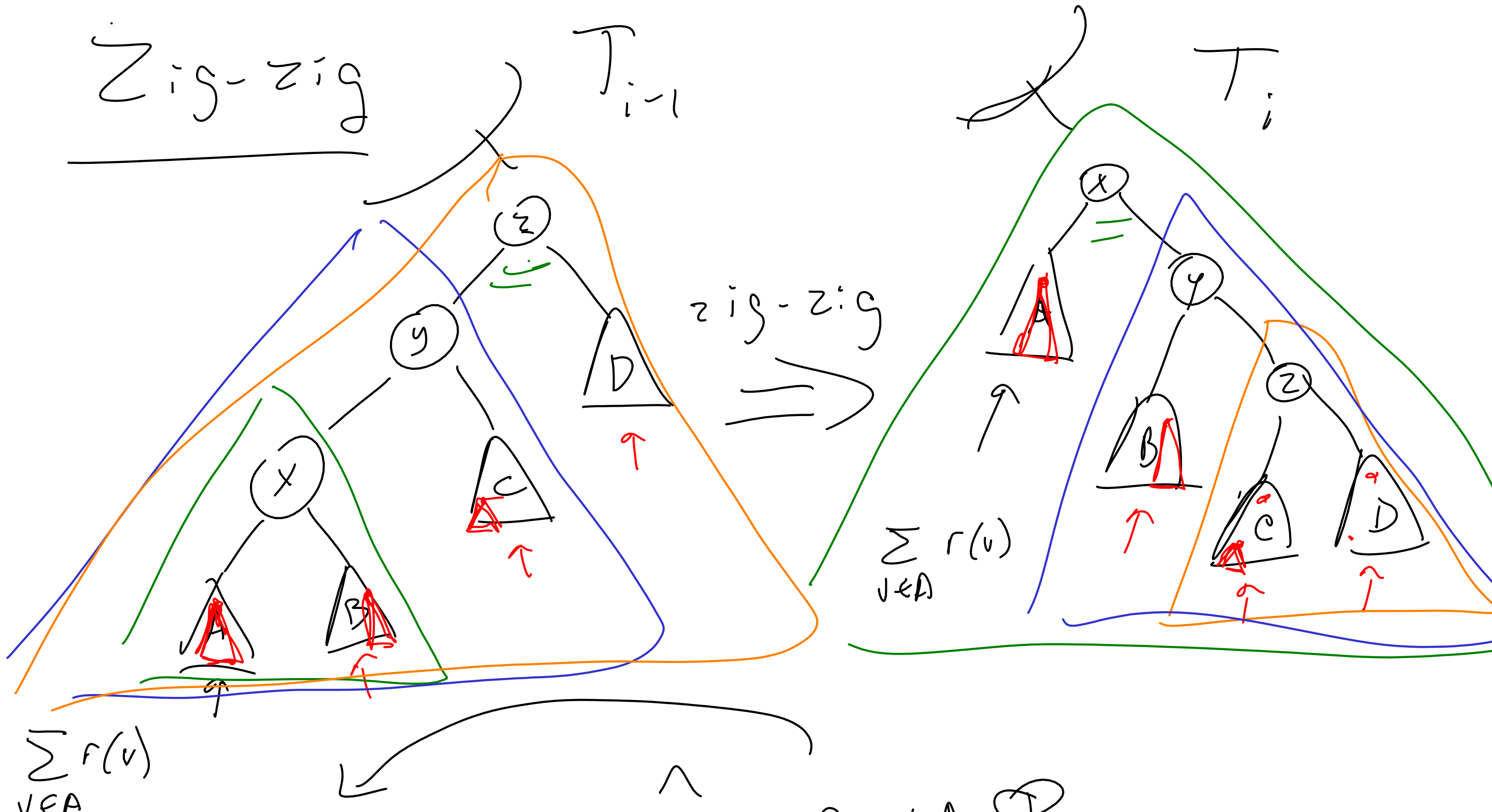
$T(x)$ ← subtree rooted at node x

$r(x) = \underline{\log s(x)}$ ← rank

$$\boxed{\Phi(T_i) = \sum_{x \in T_i} r_i(x)} = \sum_{x \in T_i} \log s(x)$$

$$= \sum_{x \in T_i} \log\Big(\sum_{v \in T(x)} w(v)\Big)$$

Zig-zig



$c_i = 2$

$\hat{c}_i = c_i + \Delta\Phi_i$

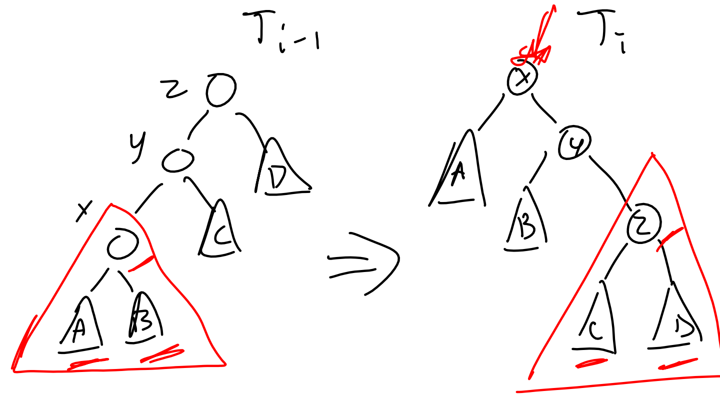
$$\Delta\Phi_i = \Phi_i - \Phi_{i-1} = \sum_{x \in T_i} r(x) - \sum_{x \in T_{i-1}} r(x)$$

$$= r_i(x) + r_i(y) + r_i(z) - r_{i-1}(x) - r_{i-1}(y) - r_{i-1}(z)$$

$$r_{i-1}(z) = r_i(x)$$

$$r_i(y) \le r_i(x)$$

$$r_{i-1}(y) \ge r_{i-1}(x)$$



$$s_{i-1}(x) + s_i(z) \le s_i(x)$$

$$\Delta\Phi_i \le r_i(x) + \cancel{r_i(x)} + r_i(z) - r_{i-1}(x) - r_{i-1}(x)$$

$$- \cancel{r_i(x)}$$

$$= r_i(x) + r_i(z) + \left[r_{i-1}(x) - r_{i-1}(x)\right] - 2r_{i-1}(x)$$

$$\log(s_i(z)) + \log(s_{i-1}(x))$$

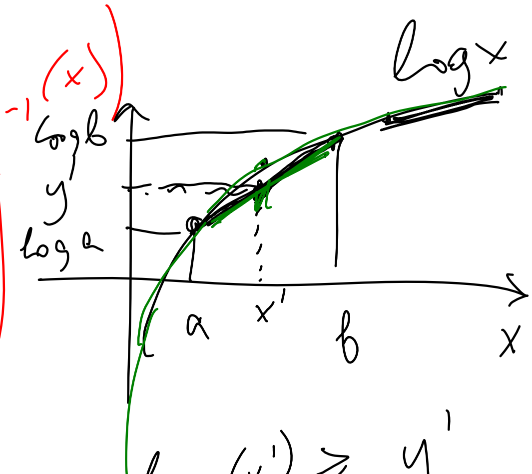
$$\le 2\log\left(\frac{s_i(z) + s_{i-1}(x)}{2}\right)$$

$$\le 2\log\frac{s_i(x)}{2}$$

$$= 2\log s_i(x) - 2\log 2$$

$$= 2\log s_i(x) - 2$$

$$= 2r_i(x) - 2$$



$$\log(x') \ge y'$$

$$x' = \frac{a+b}{2}$$

$$\log\left(\frac{a+b}{2}\right) \ge y'$$

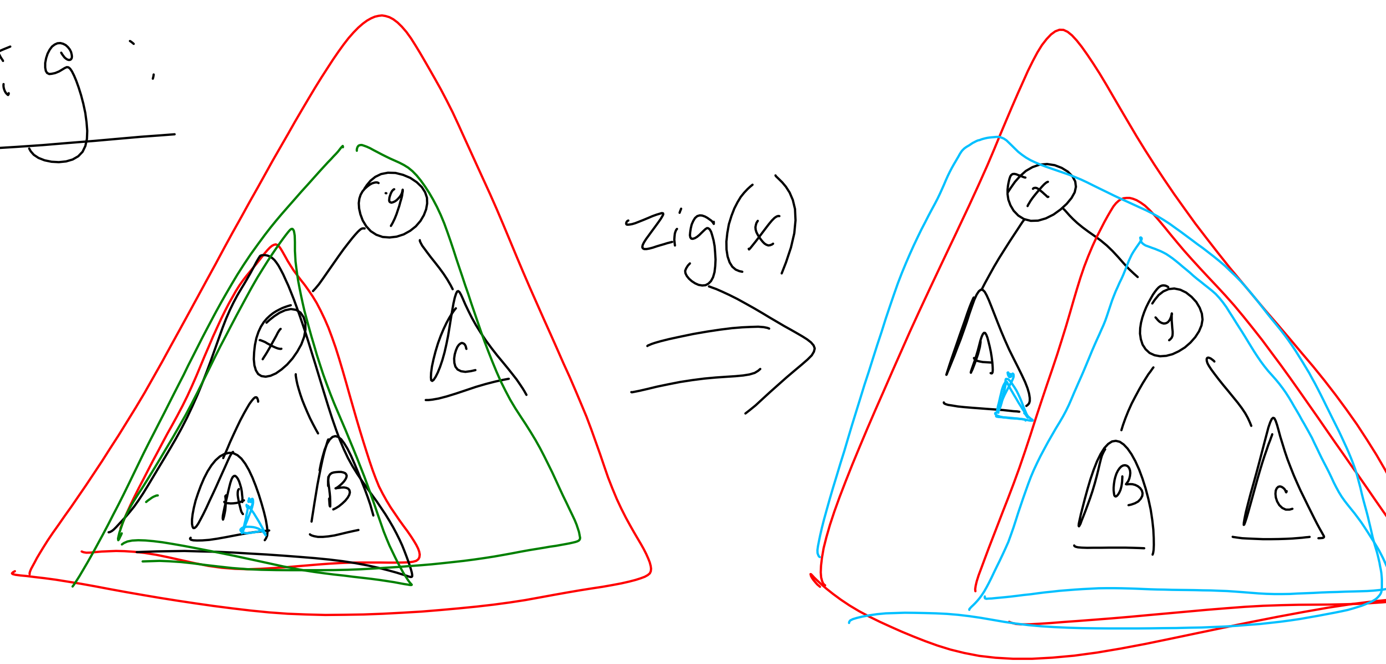
$$2\cdot\log\left(\frac{a+b}{2}\right) \ge \log a + \log b$$

$$\Delta\Phi_i \le 3r_i(x) - 3r_{i-1}(x) - 2$$

$$\hat{c}_i \le 2 + 3\left(r_i(x) - r_{i-1}(x)\right) - 2$$

$$= \underline{3\left(r_i(x) - r_{i-1}(x)\right)}$$

<u>Zig:</u>



$$\hat{c}_i = c_i + \Delta\Phi_i = 1 + \underline{\Phi_i} - \underline{\Phi_{i-1}}$$

$$= 1 + r_i(x) + \underline{r_i(y)} - r_{i-1}(x) - \underline{r_{i-1}(y)}$$

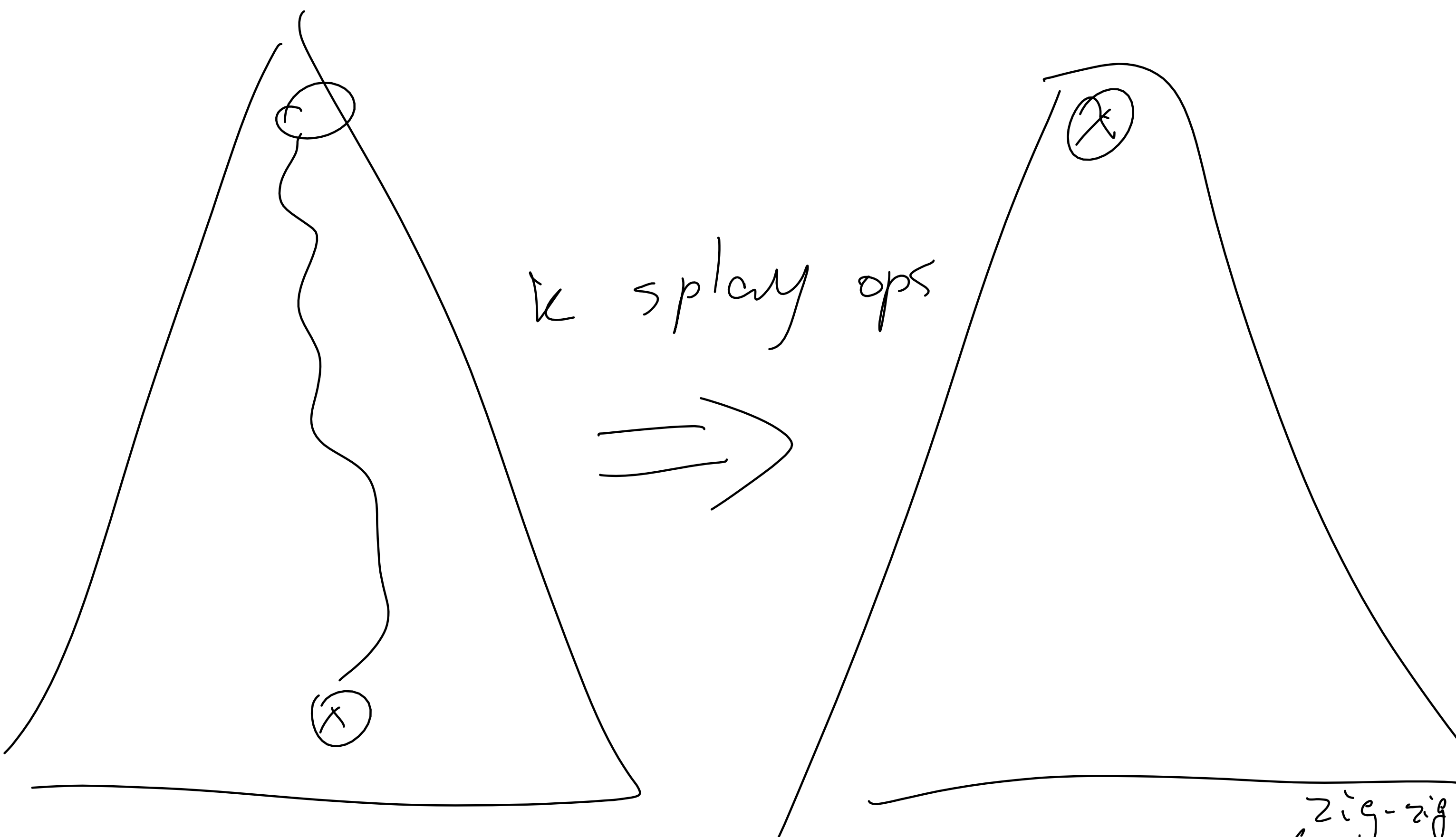
$$\leq 1 + r_i(x) + r_i(x) - r_{i-1}(x) - r_{i-1}(x)$$

$r_i(y) \leq r_i(x)$

$r_{i-1}(y) \geq r_{i-1}(x)$

$$= 1 + 2\underbrace{\left(r_i(x) - r_{i-1}(x)\right)}_{\geq 0}$$

$$\leq 1 + 3\left(r_i(x) - r_{i-1}(x)\right)$$



$$\hat{C}_{search} \leq \sum_{i=1}^{k} \hat{c}_i \leq \underset{\text{zig}}{\hat{c}_k} + \sum_{i=1}^{k-1} \hat{c}_i$$

zig-zig
zig-zag

$$\leq \underbrace{1 + 3\left(r_k(x) - r_{k-1}(x)\right)}_{\text{zig}} + \sum_{i=1}^{k-1} 3\underbrace{\left(r_i(x) - r_{i-1}(x)\right)}_{\substack{\text{zig-zig} \\ \text{or zig-zag}}}$$

$$= 1 + 3r_k(x) - 3r_0(x)$$

$$= 1 + 3\log\left(s(\text{root})\right) - 3\log s_0(x)$$

size of subtree rooted at x before the search

$$= 1 + 3\log\frac{s(\text{root})}{s_0(x)} = O\left(\log\frac{s(\text{root})}{s_0(x)}\right)$$

Amortized cost of search

$$w(x) = 1$$

$$\Rightarrow s(\text{root}) = \sum_{v \in T} w(v) = n$$

$$\hat{C}_{search} \le 1 + 3 \log \frac{n}{s_0(x)}$$

$$\le 1 + 3 \log \frac{n}{1}$$

$$\le 1 + 3 \log n$$

$$= O(\log n)$$

---

f_i — # of queries for x_i

$$\Rightarrow m = \sum_{i=1}^{n} f_i$$

$$w(x_i) = \boxed{\frac{f_i}{m} = p_i} \Rightarrow f_i = m \cdot p_i$$

$$\Rightarrow \hat{C}_Q = \sum_{i=1}^{m} \hat{c}_i \le \sum_{i=1}^{m} c \cdot \log \frac{s(root)}{s_0(x_i)}$$

$$s(root) = \sum_{x \in T} w(x) = \sum_{x_i \in T} \frac{f_i}{m} = \frac{m}{m} = 1$$

$$s_0(x_i) \geq w(x_i) = p(x_i)$$

$$\hat{c}_Q \leq \sum_{i=1}^{m} c \cdot \left(\log \frac{1}{p(x_i)} \right)$$

$$\leq c \sum_{i=1}^{n} \left(f_i \cdot \log \frac{1}{p(x_i)} \right)$$

$$= c \cdot \sum_{i=1}^{n} m \cdot p(x_i) \log \frac{1}{p(x_i)}$$

$$= c \cdot m \underbrace{\sum_{i=1}^{n} p(x_i) \cdot \log \frac{1}{p(x_i)}}_{H_n}$$

$$= c \cdot m \cdot H_n \leq c' \cdot C_{opt}$$

$$= O(m \cdot H_n)$$

$$C_{opt} = \Omega(n \cdot H_n)$$

$$C_{opt} \geq c' \cdot n H_n = \frac{c'}{c} \cdot c \cdot n H_n$$

$$\geq \frac{c'}{c} \cdot \hat{C}_Q$$

$$\hat{C}_Q \leq \frac{c'}{c} \cdot C_{opt} = c'' \cdot C_{opt}$$

$$= O(C_{opt})$$