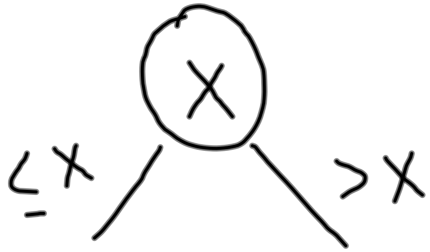
Balanced Binary Search Trees



O(lg n) height ≡ balanced

O(lg n) time Insert, Delete, Find
(Max, Min, Pred, Succ...)

Red-Black trees, 2-3 trees Scapegoat trees,
B-trees, AVL trees, ⋮

$E[\#\text{ across moves} \mid \text{list}]$

≤ 2

$E[\text{Total time}] \overset{a}{=} O(\lg n) + 2 \cdot O(\lg n)$

$= O(\lg n)$

On-line hiring

```
best = -∞
for i = 1 to k
    if score(i) > best
        best = score(i)

for i = k+1 to n
    if score(i) > best
        return i
return n
```

S = hire best applicant

$Pr(S)$?

S_i = hire best & best is i

$$Pr(S) = \sum Pr(S_i)$$

$$Pr(S_i) = 0 \quad i \leq k$$

$$Pr(S_i) = \frac{1}{n} \cdot \frac{k}{i-1} \quad i > k$$

S_i = i is best & max of the first $i-1$ is in $1..k$

$$Pr(S) = \sum_{i=k+1}^{n} \frac{k}{n(i-1)} = \frac{k}{n} \sum_{i=k}^{n-1} \frac{1}{i}$$

$$= \frac{k}{n}(\ln n - \ln k - 1)$$

Choose k to max

$k = \frac{n}{e} \Rightarrow Pr(S) = \frac{1}{e}$