Balanced Binary Search Trees

O(\log n) \text{ height } \equiv \text{ balanced}

O(\log n) \text{ time Insert, Delete, Find (Max, Min, Pred, Succ...)}

Red-Black trees, 2-3 trees, Scapegoat trees, B-trees, AVL trees, ...
\[ E(\# \text{ across mares} / 1st) \leq 2 \]

\[ E(\text{Total time}) = O(1/\sqrt{n}) + 2 \cdot O(1/g(n)) = O(1/g(n)) \]
Online 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
\[ S = \text{hire best applicant} \]
\[ \Pr(S) = \sum \Pr(S_i) \]
\[ \Pr(S_i) = 0 \quad i \leq k \]
\[ \Pr(S_i) = \frac{1}{n} \cdot \frac{k}{i-1} \quad i > k \]
\[ S_i = i \text{ is best and max of the first } i - 1 \text{ is in } 1 \ldots k \]
\[ R(S) = \sum \frac{k}{i} \cdot \frac{1}{\eta(i-1)} - \frac{k}{i} \leq \frac{1}{i} \]
\[ \text{Choose } k \text{ to } \max \chi \]
\[ k = \frac{n}{e} \Rightarrow \Pr(S) = \frac{1}{e} \]
\[ k = \frac{n}{e} \Rightarrow \Pr(S) = \frac{1}{e} \]