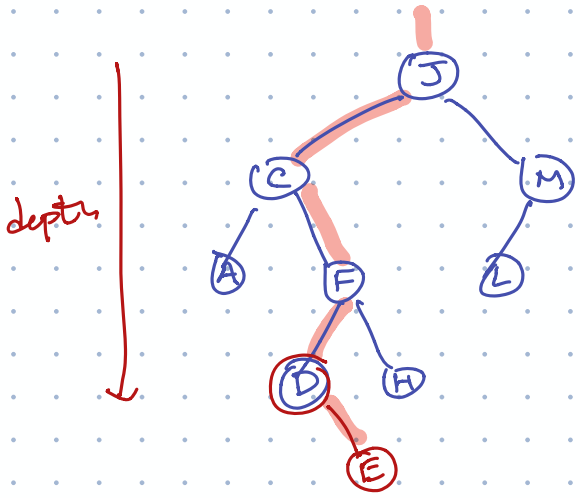
# Binary Search Tree




Find(x) — Pred(x), Succ(x)

Insert(x)

Delete(x)

$T_1, T_2 \leftarrow$ Split(x)

$T \leftarrow$ Join($T_1, T_2$)

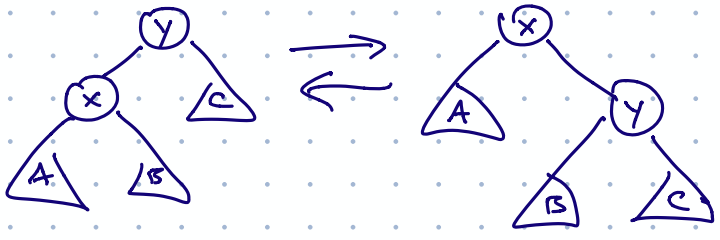
BSTs have depth $O(n)$ — we want $O(\log n)$

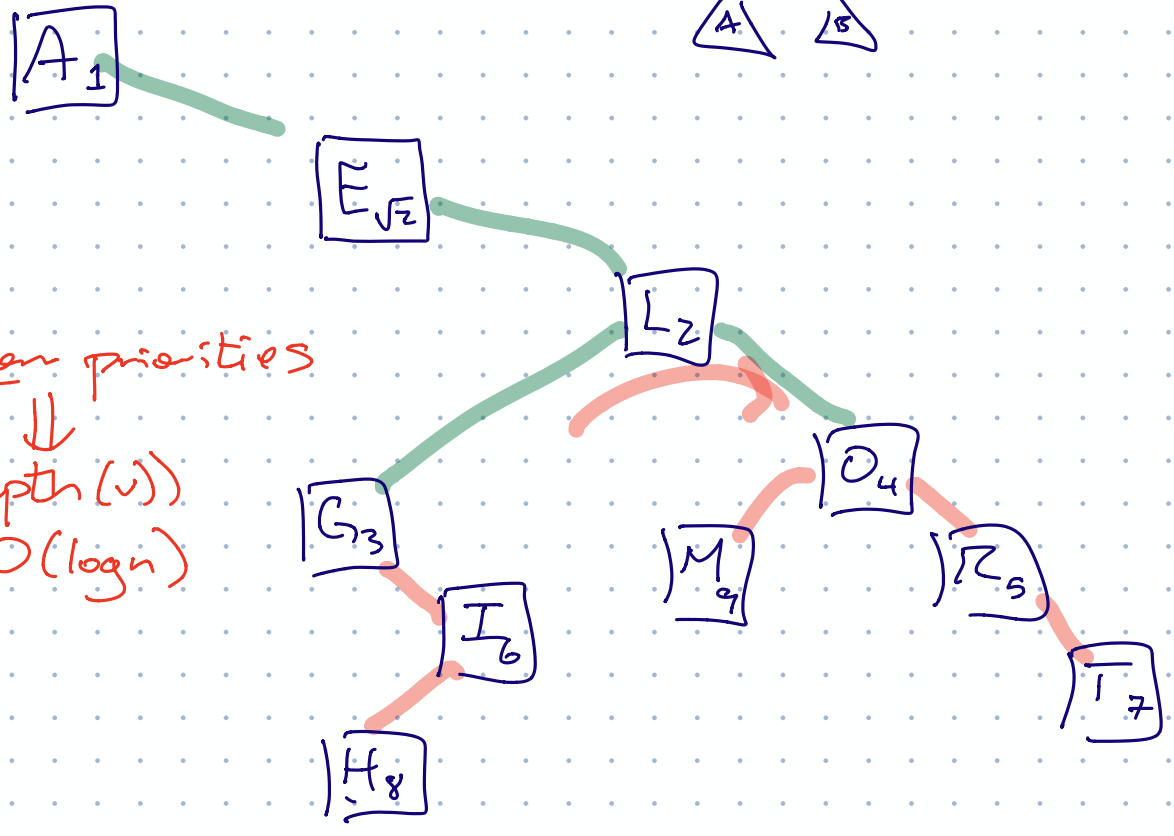
---

**Treap** — binary search **tree** × min-**heap**

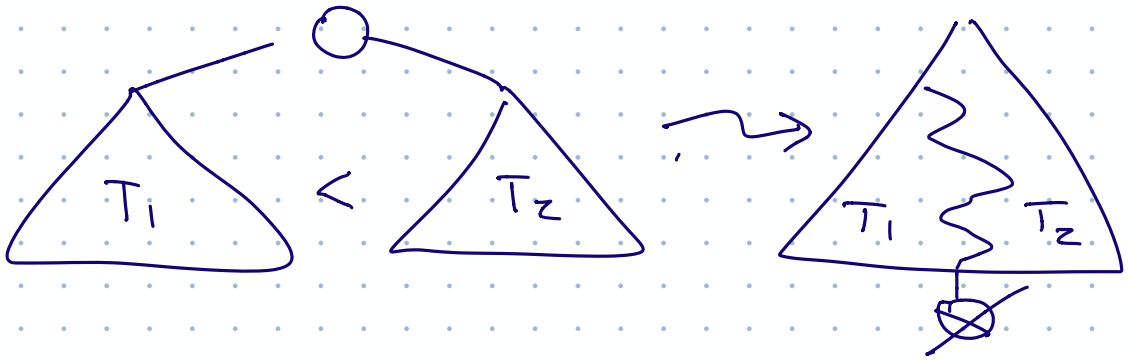
Binary tree
BST for keys
and min-heap
for priorities


letter = key
number = priority




Random priorities
⇓
$E(\text{depth}(v)) = O(\log n)$

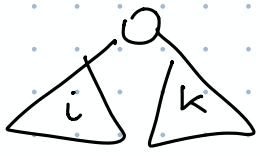



$$E[\text{depth}(k)] = E[\#\text{ proper ancestors of } k]$$

$$= \sum_{i=1}^{n} \Pr[i \uparrow k] \qquad i \uparrow k = i \text{ is a proper ancestor of } k.$$

**Lemma:** $i \uparrow k$ iff $\text{priority}(i) = \min\{\text{priority}(j) \mid i \le j \le k\}$ for all $i < k$

**Proof:** Several cases:

- root $= i$ — $i \uparrow k$ — priority$(i)$ is min
- root $= k$ — $i \not\uparrow k$ — priority$(i)$ is not min
- root $< i$ — $i$ and $k$ in right subtree — IH ✓
- $i <$ root $< k$ —  — $i \not\uparrow k$ — priority$(i)$ is not min
- root $> k$ — $i$ and $k$ in left subtree — IH ✓

□



$$\Pr[i \uparrow k] = \frac{[k \neq i]}{|k - i| + 1}$$

$$E[\text{depth}(k)] = \sum_{i=1}^{n} \Pr[i \uparrow k]$$

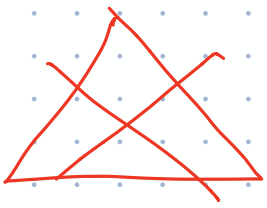
$$= \sum_{i<k} \Pr[i \uparrow k] + \Pr[k \uparrow k] + \sum_{i>k} \Pr[i \uparrow k]$$

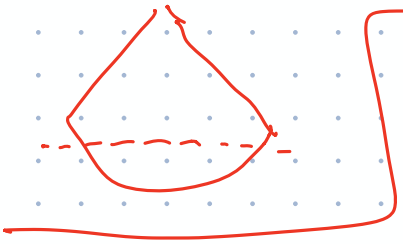
$$= \sum_{i=1}^{k-1} \frac{1}{k - i + 1} + \sum_{i \ge k+1}^{n} \frac{1}{i - k + 1}$$

$j = k-i+1$ $\qquad$ $j = i-k+1$

$$\sum_{j=2}^{k} \frac{1}{j} + \sum_{j=2}^{n-k+1} \frac{1}{j}$$

$$H_k - 1 + H_{n-k+1} - 1 \leq 2 \ln n - 2$$





Treap = BST obtained by inserting keys in increasing priority order

Treap = recursion tree for randomized quicksort

pivot = min priority

RANDOM QUICKSORT(A[1..n]):

```
T ← empty BST
for i ← 1 to n in random order
    Insert(A[i])
return inorder(T)
```

---

For all $k$, $E[\text{depth}(k)] = O(\log n)$

Not yet: $E[\max_k \text{depth}(k)] = O(\log n)$