Binary Search Trees

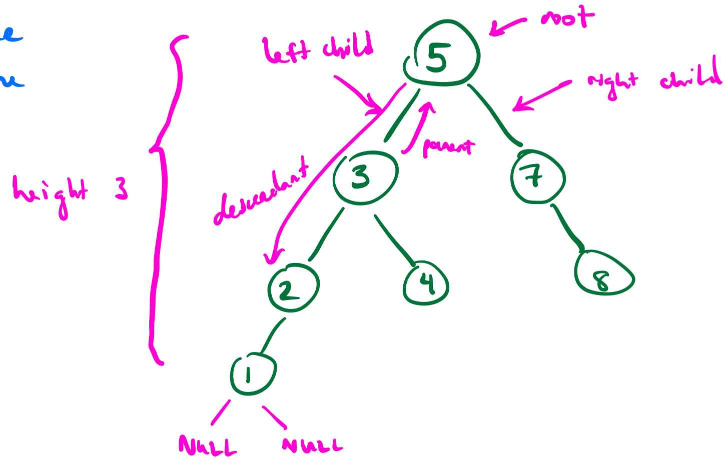
Motivation: for storing numbers / sortable values:

Sorted arrays: search $O(\log n)$
insert $O(n)$
delete $O(n)$

Linked lists: search $O(n)$
insert $O(1)$
delete $O(n)$

Binary search trees: search $O(\log n)$
(when balanced) insert $O(\log n)$
delete $O(\log n)$

Binary tree structure



Binary search trees

for any parent node:

→ all left descendants have keys less than parent
→ all right descendants have keys greater than parent

In-order traversal:

```
traverse (x):
    if x:
        traverse (x.left)
        print (x.key)
        traverse (x.right)
```

$O(n)$ time

Search

```
search (x, parent, key)
    if !x:
        return parent
    if x.key == key:
        return x
    if x.key > key:
        return search (x.left, parent, key)
    else:
        return search (x.right, parent, key)
```

$O(\log n)$ time ($O(n)$ if imbalanced)

Insert

```
insert (key):
    x = search (key)
    if key < x.key:
        x.left = key
    else:
        x.right = key
```

$O(T(\text{search}))$ time

Delete

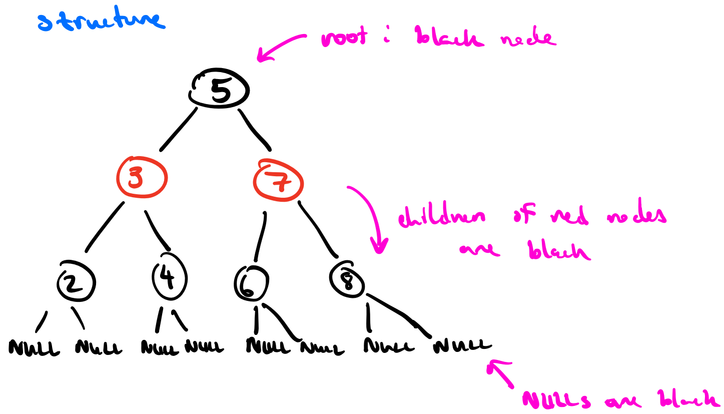
```
delete (key):
    x = search (key)
    if x.key == key:
        delete key (separate cases for leaf vs internal nodes)
```

$O(T(\text{search}))$ time

Red-Black Trees

→ self-balancing BST: ensures black nodes are balanced
+ not too many red nodes.

structure



For all nodes x, all paths from x to NULLs have same # black nodes on them.

→ rules: proxy for balance.

→ can maintain using rotations on insert/delete

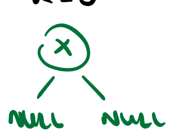
RB-tree height theorem

Thm. Height of an RB-Tree with n non-NULL nodes is at most $2\log(n+1)$

Claim: at least $2^{b(x)}-1$ non-NULL nodes in the subtree under x, including x, where $b(x)$ is # black nodes in any path from x to NULL

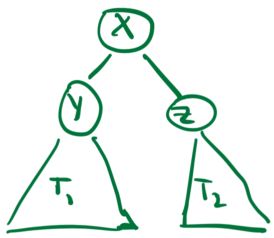
Ind. hyp.: Claim holds for trees of ht $\leq k$

Base case: $k=0$



$1 \geq 2^{1}-1$

Ind. step: Say IH holds for $k = t-1$, show for $k = t$:



$b(y) = b(x)$ or $b(x) - 1$

if y is red   if y is black

$\Rightarrow \; b(y) \geq b(x) - 1$

$b(z) \geq b(x) - 1$

#nodes $\geq 1 +$ (#nodes in T_1) $+$ (#nodes in T_2) $\geq$

$$1 + (2^{b(y)} - 1) + (2^{b(z)} - 1)$$

$$\geq 1 + (2^{b(x)-1} - 1) + (2^{b(x)-1} - 1)$$

$$= 2^{b(x)} - 1$$

$$\Rightarrow \; n \geq 2^{b(root)} - 1$$

$$\geq 2^{height/2} - 1$$

$$\Rightarrow \; n+1 \geq 2^{height/2}$$

$$\Rightarrow \; height \leq 2\log(n+1)$$