**Trees**

**Binary Tree: Find(val)**

How would you search for an element in a generic binary tree?

- Walk the tree
  - inorder
  - preorder
  - postorder

- What's the worst case?

- What's the best case?

- Average? $\rightarrow$ Also $n$ ($\frac{n}{2}$ on average)
Binary Tree: Find Min

How would you search for the smallest element in a generic binary tree?

Tree structure doesn’t matter!
Same time as using a flat list.
- Best: $O(n)$
- Avg: $O(n)$
- Worst: $O(n)$
Binary Search Tree: Find Min

How would you search for the smallest element in a binary search tree?

- What's the worst case?

- What's the best case?

---

- Average?

We could show (but won’t) that the expected height of a binary search tree built with equally likely data orderings is $O(\log n)$. 

Not necessarily balanced.
**Binary Search Tree: Find Next Largest.**

How would you find the next largest element in a binary search tree based on the element at which you are currently positioned? Again, not necessarily balanced.

**Cases:**

If there is a right subtree, it is in there - find smallest element in left most leaf node within this subtree.

**Example:**

```
    10
   / \
  5   12
 /   /   \
2    7    18
```

**Case 2:** Otherwise, follow path upward until an ancestor is found that is greater than the current element. (Might be the root node.)

**Example:**

```
    10
   /   \
  5    12
 /     /   \
2      7    11
```

Next time: Work on height.

$O(\text{height})$