

# Discussion 05

## Trees

Aditya Balasubramanian

`aditbala [at] berkeley [dot] edu`

# Announcements

- Project 2 (C.A.T.S.) is due Friday 2/24.
  - Early submission bonus point for finishing by Thursday 2/23.

# Trees

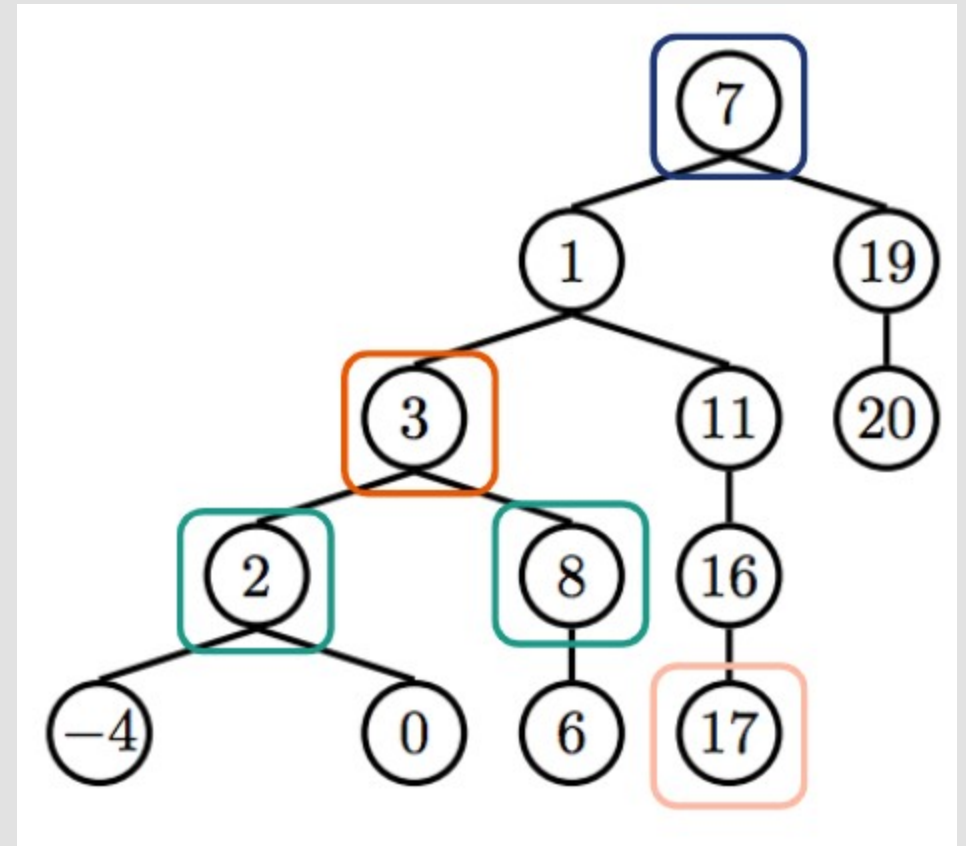


# Tree

- What are they?
  - Data structure for hierarchies of data
- What should we know
  - Recursion!
  - Every subtree is also a Tree

# Tree Terminology

- Parent Node
  - A node that has branches
- Child Node
  - A node with a parent
  - Can only have one parent
- Root
  - The top node in a tree
  - There is only one root for a tree
- Label
  - The value of a node



# More Tree Terminology

- Leaf
  - A node with no branches
- Branch
  - A subtree of the root
  - All branches are also trees
- Depth
  - How far away a node is from the root
- Height
  - The depth of the lowest leaf



# ADT **tree** Implementation

```
def tree(label, branches=[]):  
    """Construct a tree with the given label value and a list of branches."""  
    return [label] + list(branches)  
  
def label(tree):  
    """Return the label value of a tree."""  
    return tree[0]  
  
def branches(tree):  
    """Return the list of branches of the given tree."""  
    return tree[1:]  
  
def is_leaf(tree):  
    """Returns True if the given tree's list of branches is empty, and False  
    otherwise.  
    """  
    return not branches(tree)
```

# tree

- Initializing a `Tree`
  - `tree(label, branches=[])`
  - `t = tree(1, [tree(2), tree(3)])`
- Accessing branches of a `tree`
  - `branches(t) -> [tree(2), tree(3)]`
- Checking if a `tree` is a leaf
  - `is_leaf(t) -> False`
  - `is_leaf(branches(t)[0]) -> True`
- Getting label of a `Tree`
  - `label(t) -> 1`



# Manipulating Trees

```
for b in branches(t)
```

- What is this?
  - IMPORTANT line for dealing with a `tree`
- Why use this?
  - Allows us to iterate through branches of a `tree`
  - Useful for calling recursive functions on all branches of a `tree`
- Can also be a base case
  - The for loop does not run if there are no branches to iterate over

# How do I use this? (Recursion for trees)

## 1. Base Case

- Smallest Input
- Usually a leaf

## 2. Recursive Calls

- Call recursive function on branches

## 3. Putting it together

- Use recursive calls to solve problem
- Can use `max`, `min`, `sum`, `any`, `all` on lists

# Thank you

Anon Feedback -> <https://tinyurl.com/adit-anon>