



Computational Structures in Data Science



UC Berkeley EECS
Lecturer Michael Ball

Lecture #19: Data Structures: Trees



Updates

- Updated declaration policies
 - <https://piazza.com/class/hyq0br1u3kx7dg?cid=11615>
 - <Https://bit.ly/eecs-piazza> to sign up for “EECS 101”
- If you need help, please reach out.
- Please checkout the midterm survey
 - <https://piazza.com/class/k5kga9pxw0l754?cid=666>
- As a reminder: Private piazza posts are best, since all the staff see them.



Computing In The News

How a Real Dog Taught a Robot Dog to Walk

“Instead of coding a mechanical quadruped's movements line by line, Google researchers fed it videos of real-life pups. Now it can even chase its tail.”

<https://www.wired.com/story/how-a-real-dog-taught-a-robot-dog-to-walk/>





Why?

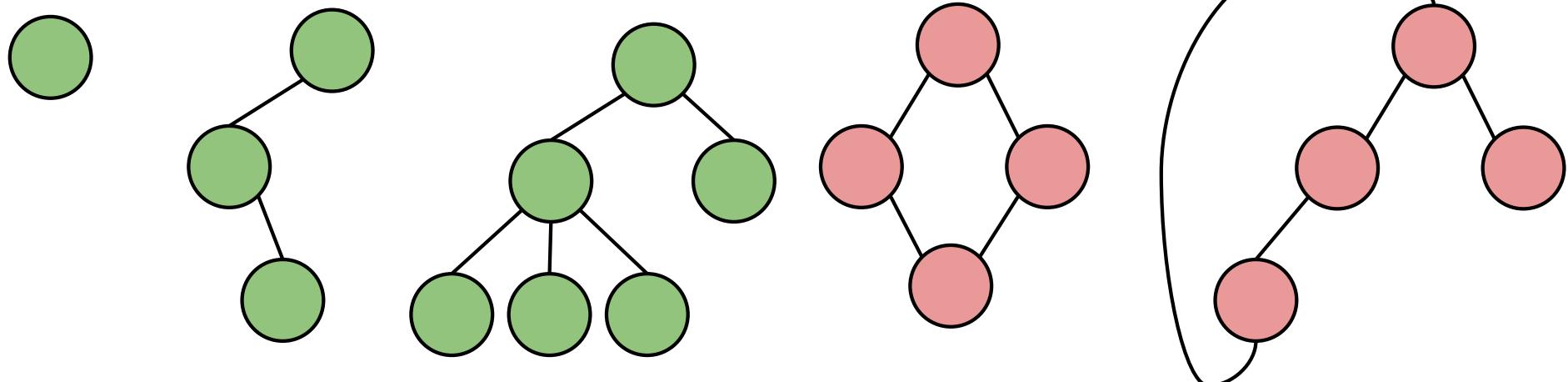
- **Trees represent lots of natural structures**
 - A boss who has employees report to them
 - Courses which belong to departments, and departments which colleges in a University
 - Anything with a hierarchy, really.
 - » A family tree
 - » Biological taxonomies (Kingdom, Phylum....)
- **Trees give us really cool approaches for “divide and conquer”**
 - Used in every computer to speed up searching for files
- **Another recursive data structure!**
 - We can keep practicing recursion and working with classes
- **Trees are a simplified form of a *graph*, a tool which can help us model just about anything.**



What's a tree?

- A set of nodes
- A set of edges or branches
 - Constraint: There is exactly one path between any two nodes

Green structures below are trees. Pink ones are not (right two). A tree with more than one path is a graph...we won't talk about those in CS88.



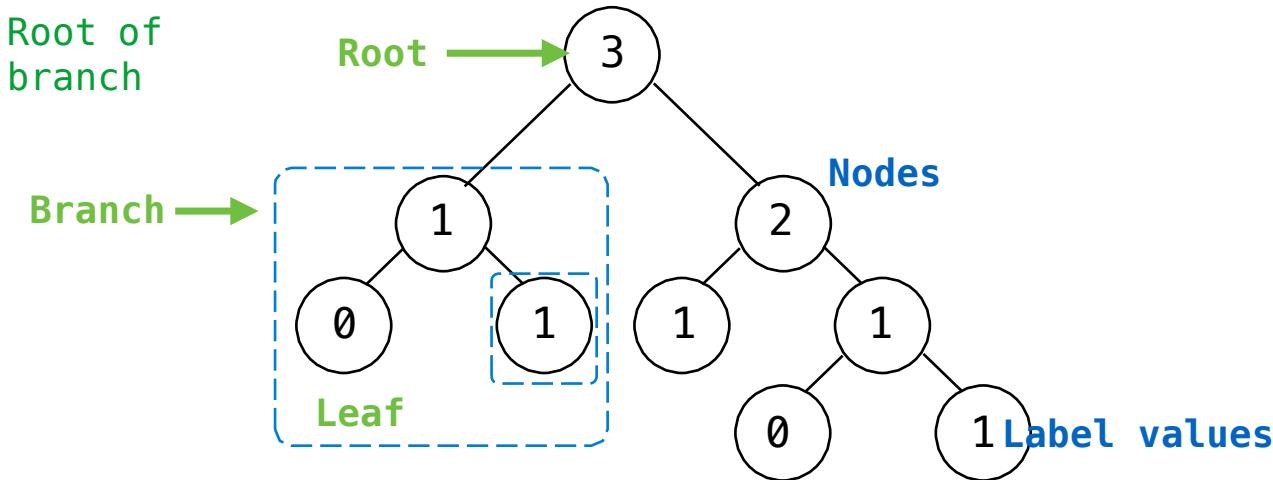


Some Useful Terms

- **Entry or Node**
 - An item in a tree, it is also a Tree
 - Each one has a value.
- **Root**
 - The very first item at the top of a tree
- **Edges, Branches**
 - An implicit connection from one entry to the next



Thinking About Trees



Recursive description (wooden trees):

A **tree** has a **root** and a list of **branches**

Each branch is a **tree**

A tree with zero branches is called a **leaf**

Relative description (family trees):

Each location in a tree is called a **node**

Each **node** has a **label value**

One node can be the **parent/child** of another

People often refer to values by their locations: "each parent is the sum of its children"



Tree Recursion

- **Fib(4) → 9 Calls**
- **Fib(5) → 16 Calls**
- **Fib(6) → 26 Calls**
- **Fib(7) → 43 Calls**
- **Fib(20) →**

