


UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science



Data Structures: Trees

UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

1

Computing In The News: Go Bears!




- Training Facial Recognition on Some New Furry Friends: Bears ([NYTimes](#))
- "In their spare time, two Silicon Valley developers aided conservationists in developing artificial intelligence to help keep track of individual bears."
- Annotated thousands of photos by hand to start their work!



UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

2

Updates: Ants




- Ants checkpoint is due Friday night
- Remember Ants is a partner project!
- "live" lecture Wednesday – only 2 left!
- Next week: No class Weds, Thurs, Friday.
- Extended self-checks by a couple days for this lecture.

UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

3

UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science




Data Structures: Trees

UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

4

Learning Objectives




- Trees are a general version of linked lists
- Trees have a value, and are connected to "sub-trees" called branches
- We can often use recursion to process all items in a tree
 - We typically have recursion inside a loop over all of the tree's branches

UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

5

Why Learn Trees?



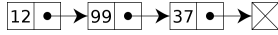
- Trees represent lots of natural structures
 - A boss who has employees report to them
 - Courses which belong to departments, and departments which colleges in a University
 - Anything with a hierarchy, really.
 - » A family tree
 - » Biological taxonomies (Kingdom, Phylum...)

UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

6

Review: Linked Lists

- A Recursive List, sometimes called a "rlist"
- Linked lists contain other linked lists
- A series of items with two pieces:
 - A value, usually called "first"
 - A "pointer" to the rest of the items in the list.



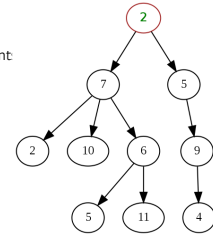
- We'll use a very small Python class "Link" to model this.

UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

7

What is a tree?

- A recursive data structure
 - Almost like a linked list!
- What if a linked list could have multiple "rest" element?
- We call these "branches".
- **Each branch is also its own Tree.**



UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

8

Trees are common in Computer Science

- Trees give us really cool approaches for "divide and conquer"
 - Used in every computer to speed up searching for files
 - Used for modeling decision systems in AI programs
 - Used for modelling the kinds of moves in a game.
- Another recursive data structure!
 - We can keep practicing recursion and working with classes
 - Computer sciences really like recursion. ☺
- Trees are a simplified form of a *graph*, a tool which can help us model just about anything.

UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

9

Computational Structures in Data Science



UC Berkeley EECS
Lecturer
Michael Ball

Trees: Code Overview

UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

10

Computational Structures in Data Science



UC Berkeley EECS
Lecturer
Michael Ball

Trees: Adding And Inspecting Branches

UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

11

Computational Structures in Data Science




UC Berkeley EECS
Lecturer
Michael Ball

Trees: Traversing Each Node


UC Berkeley | Computer Science 98 | Michael Ball | <http://cs98.org>

12



UC Berkeley EECS
Lecturer
Michael Ball


Computational Structures in Data Science



Trees: Practice With Recursion


UC Berkeley | Computer Science 68 | Michael Ball | <http://cs68.org>

13



UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science




Trees: Advanced Topics: Searching Optional!

UC Berkeley | Computer Science 68 | Michael Ball | <http://cs68.org>

14

Searching Trees: Two Strategies




- The searching we have been doing today is called “Depth First Search”, or DFS.
- Recursion makes the algorithm very nice.
 - First: we deal with our current item, then we get to the branches.
 - We always make a recursive call on the first branch
 - We continue recursing until there are no more branches
 - Then the function executes, and we go back “up” a level and check out the next branch.
 - We sometimes say: “popping up the stack”.
 - The *stack* is the “stack of function calls” the computer uses to keep track of how things work, and you’ll learn about this in CS61B.

UC Berkeley | Computer Science 68 | Michael Ball | <http://cs68.org>

15

Searching a Tree by level: Breadth First Search



- What if I want to check out all the values of my branches before making a recursive call?
- What if we said, you just can’t use recursion. (Sometimes, CS instructors do weird things like that...)
- This is used in practice for lots of cool things:
 - Shortest path between two items (more of a graph and not a tree, usually). Google Maps uses it for routing and the algorithms that power the internet use it.

UC Berkeley | Computer Science 68 | Michael Ball | <http://cs68.org>

16