Discussion 5

Data Abstraction, Trees

Antonio Kam

anto [at] berkeley [dot] edu

Announcements

- Cats due Friday
  - but also due tomorrow because EC point
  - please try to get it
- I'd highly recommend giving the lab a try if you haven't:
  - flatten (very good tree recursion question! (it's hard))
  - q 5, 6, 7
- You still need to submit something to Gradescope to get credit!!

Comments from last section

- Switch on the lights at the front
- Apples with cheddar cheese
  - i have never tried it, but this actually sounds really hype
  - i love cheese
- favourite boba drink/tea
  - chrysanthemum (honey) tea is 👌
- test problems
  - usually don't have the bandwidth for this during discussion, but when i find things very important, i will go for it

Temperature Check 🌡️

- Recursion
- Tree Recursion
- Data Abstractions
- Trees! 🎄

All slides can be found on

`teaching.rouxl.es`

Data Abstractions

What are Data Abstractions?

- Data abstractions are a super powerful way to let people treat code as objects, rather than knowing how the thing works itself
- Allows you to worry about how something works, rather than how something is implemented
- You'll see a lot of abstractions in other courses (Data 8, Data 100 are filled with abstractions of some sort)

What are Data Abstractions?

- Data abstractions have the following:
  - Constructors: Used to build the abstract data type
    - IMPORTANT: You do not need to know how the programmer decided to implement this!
  - Selectors: Used to interact with the data type

Example: Tree Data Abstraction

- Trees are recursive data structures (as in, trees contain more trees)
- Important terms:
  - Root Node
  - Branch(es)
    - This will be a list!
  - Leaf Node
  - Children
- Sort of looks like an upside-down tree compared to the real world
- Questions are generally solved using tree recursions

Tree ADT Implementation:

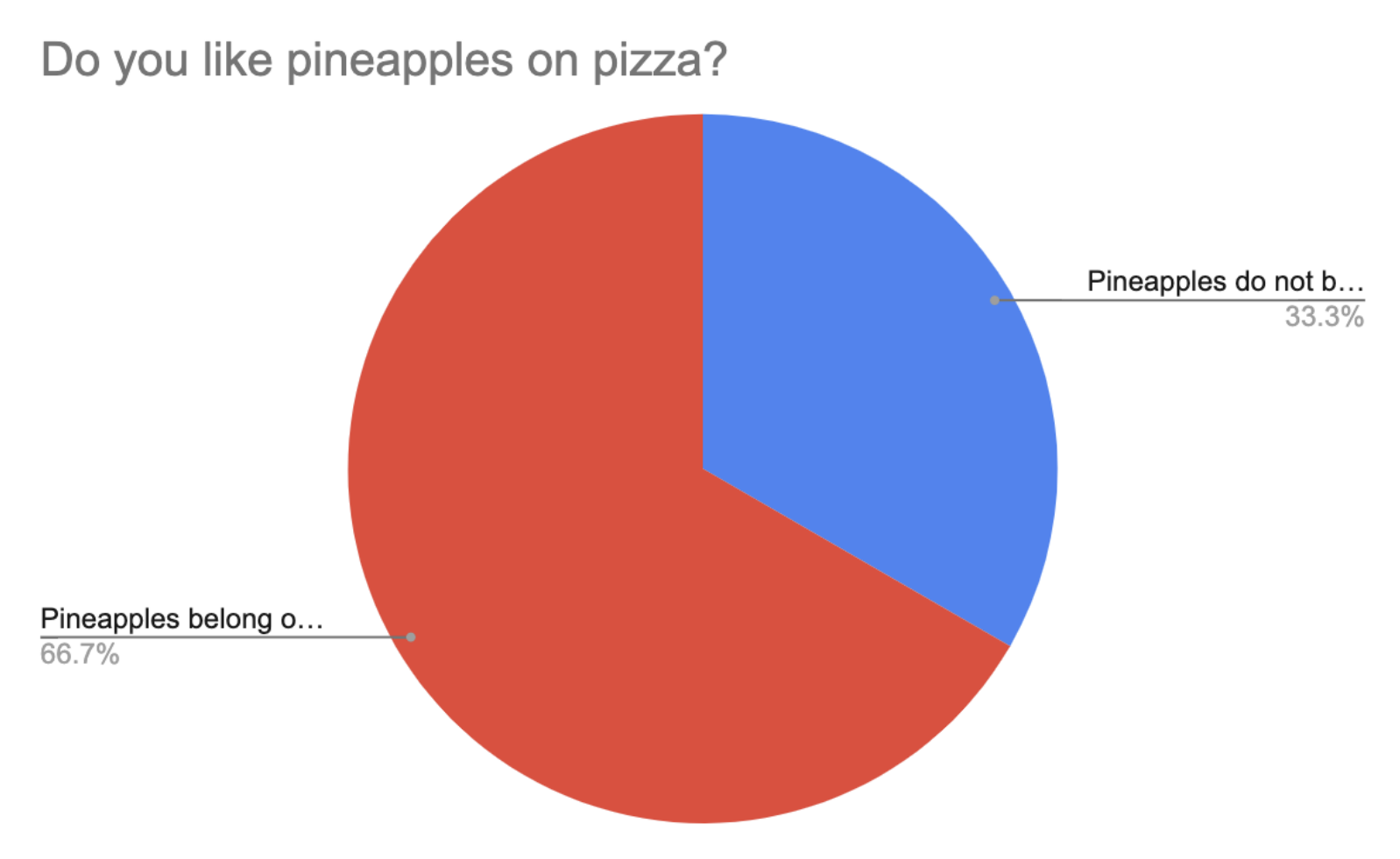
```
def tree(label, branches=[]):
    """Construct a tree with the given label value and a list of branches."""
    return [label] + list(branches) # All items in branches must be trees!

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):
    return not branches(tree)
```

Tree Example:

```
t = tree(1,
      [tree(3,
          [tree(4),
           tree(5),
           tree(6)]),
       tree(2)])
```

Worksheet!

Results from last section (`links.rouxl.es/disc`)



Mental Health Resources

- CAPS:
  - If you need to talk to a professional, please call CAPS at 510-642-9494.
- After Hours Assistance
  - For any assistance after hours, details on what to do can be found at this link

Anonymous Feedback Form

`links.rouxl.es/feedback`

Thanks for coming! 🥳

Please give me feedback on what to improve!