We’re having a spring mountain day.

NO LAB THIS WEEK

Go outside and enjoy the spring weather.
NO LAB THIS WEEK

I’ll be in my office or the lab during Monday’s regularly scheduled lab hours to answer questions, go over homework problems, etc. come say hi.
Welcome to CS 134!

Introduction to Computer Science
Iris Howley

- Binary Trees II -
TREES

A data structure for sorting and searching
Is it alive?

Does it have 8 legs?

Does it have 4 legs?

Is it an octopus?

Is it sweet?

Is it a pretzel?

Is it a table?

Is it food?

no

yes

no

yes

no

yes
Twenty Questions Tree

```python
__slots__ = [value, left, right]
```

```
Is it alive?
  yes
  Does it have 8 legs?
    yes
    Is it an octopus?
      yes
      Is it a pretzel?
        yes
        Is it a table?
          no
    no
    Is it food?
      yes
      Is it sweet?
        yes
        Does it have 4 legs?
          yes
          Is it a table?
```

no
no
yes
yes
no
Binary Tree

- Let’s write a `contains(..)` method for a tree
  - (Application Question #3 from POGIL #40)

```python
>>> from tree import *
>>> mytree = tree(99, tree(33), tree(66))
>>> 66 in mytree
True  # __contains__() is implicitly called with "if ___ in <sequence>"
>>> 24 in mytree
False
```

(this will be helpful for completing your homework)
Contains Method for tree

• Stopping/Base Case:
  1. We’ve found the value
  2. Or we’re a leaf!

• Small step
  ▪ Check if we’re the value

• Break the journey down
  ▪ Check the left child, then the right (if it’s not in the left side)
Contains Method for tree

# __contains__() is implicitly called with “if ___ in <sequence>”

- def __contains__(self, v):
  - # Base case
  - if self.value == v:
    - return True
  - l = v in self.left if self.left else False
  - r = v in self.right if not l and self.right else False
    # if not l lets us skip the right side, if we found it in the left already
  - return l or r
Twenty Questions Tree

```python
__slots__ = [\_value, \_left, \_right]
```

Questions stored as the value

- Is it alive?
  - yes
  - Does it have 8 legs?
    - yes
      - Is it an octopus?
        - yes
        - Is it a pretzel?
          - no
          - Is it a table?
            - yes
    - no
      - Is it food?
        - yes
        - Is it sweet?
          - yes
        - Does it have 4 legs?
          - no

If it’s a leaf, it’s a guess

‘yes’ goes to the left
‘no’ to the right
q20.py

A program to play 20 Questions, using our tree data structure

See shared/examples/04.22
NO LAB THIS WEEK

I’ll be in my office or the lab during Monday’s regularly scheduled lab hours to answer questions, go over homework problems, etc.

come say hi.
QUESTIONS?
Leftover Slides
Binary Tree Data Structure

• POGIL Activity #20 (Renamed #40) goes over the specifics of the Binary Tree data structure

• shared/examples/04.19 has the code for Binary Tree
Creating a Tree

• t2 = Tree('Does it have 8 legs?')
• t3 = Tree('Is it food?')
• mytree = Tree('Is it alive?', t2, t3)
Adding Nodes to the Tree

• Octopus?

• $t_4 = (\text{"octopus?")}$

• $\rightarrow t_2 = \text{Tree(`8 legs?',T4)}$
Accessing Nodes in a Tree

- `print(mytree.value)` 'Is it alive?'
- `print(mytree.left.value)` 'Does it have 8 legs?'
- `print(mytree.left.left.value)` 'Is it an octopus?'
What does this code do?

```python
• def mystery(self):
  •     if not self.right:
  •         return self

  •     return self.right.mystery()
```
See the rest of the POGIL Activity...

- Write a recursive method of Tree that returns the left most leaf of a Tree. In 20 Questions, that would be the “Is it an octopus?” node.
- `def leftmost(self):`