Twenty Questions
(or: How your next homework works)
(or: Trees!)
I’m thinking of a noun.
One more time!
Is it bigger than a breadbox?

No
Is it something you find indoors?

Yes
Is it alive?
Is it bigger than a breadbox?
   No  
   Is it something you find indoors?
   Yes  
   Is it alive?
   No  
   Is it alive?
   Yes  
   Is it something you move often?
Trees are more than just for Sets & Maps!
Is it bigger than a breadbox?

Yes

No

Is it something you find indoors?

No

Is it alive?

Yes

Is it a solid?

No

Is it an animal?

Yes

Can you eat it?

No

Is it blue?

Yes

No

Is it alive?

Yes

No

Is it something you move often?
Is it bigger than a breadbox?

- Yes
  - Is it alive?
    - Yes
      - Is it blue?
        - Yes
          - Can you eat it?
            - No
          - Yes
            - It's a breadbox!
    - No
      - Is it alive?
        - Yes
          - Is it blue?
            - Yes
              - It's a breadbox!
            - No
              - It's a breadbox!
        - No
          - It's a breadbox!

- No
  - Is it something you find indoors?
    - Yes
      - Is it a solid?
        - Yes
          - Is it an animal?
            - Yes
              - Can you eat it?
                - No
              - It's a breadbox!
            - No
              - Is it blue?
                - Yes
                  - Can you eat it?
                    - No
                  - It's a breadbox!
                - No
                  - It's a breadbox!
          - No
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
        - No
          - Is it alive?
            - Yes
              - Is it blue?
                - Yes
                  - Can you eat it?
                    - No
                  - It's a breadbox!
                - No
                  - It's a breadbox!
            - No
              - It's a breadbox!
      - No
        - Is it alive?
          - Yes
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
            - No
              - It's a breadbox!
        - No
          - It's a breadbox!
  - No
    - Is it something you move often?
      - Yes
        - Is it alive?
          - Yes
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
          - No
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
      - No
        - Is it alive?
          - Yes
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
          - No
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
      - No
        - Is it alive?
          - Yes
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
          - No
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
      - No
        - Is it alive?
          - Yes
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
          - No
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
      - No
        - Is it alive?
          - Yes
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
          - No
            - Is it blue?
              - Yes
                - Can you eat it?
                  - No
                - It's a breadbox!
              - No
                - It's a breadbox!
      - No
        - Is it alive?
Grows below ground?

No

Goes in sandwiches?

Yes

Often in salads?

Wednesday, October 24, 12
What do we need to store here?

Grows below ground?

No

Goes in sandwiches?

Yes

Often in salads?

Or here?

Yes

No
Learning Twenty Questions

(below ground)

Often in salads?

No

Yes
Learning Twenty Questions

(below ground)

Often in salads?

No

Yes

Wednesday, October 24, 12
Learning Twenty Questions

(below ground)

Often in salads?

No

Yes

This is a ginger root.
Learning Twenty Questions

(below ground)

Often in salads?

No

Yes

Is it commonly mashed?
Learning Twenty Questions

Often in salads?
- Yes
- No (below ground)

Is it commonly mashed?
- No (below ground)
- Yes

Wednesday, October 24, 12
Saving Twenty Questions
// Suppose we have a ListNode class...
public void saveList(ListNode n) {
    if (n == null) {
        return; \textit{Base case}
    }
    System.out.println(n.getValue()); \textit{Deal with this node}
    saveList(n.getNext()); \textit{Recurse}
Saving Twenty Questions

// Suppose we have a ListNode class...
public void saveList(ListNode n) {
    if (n == null) {
        return; // Base case
    }
    System.out.println(n.getValue()); // Deal with this node
    saveList(n.getNext()); // Recurse
}

// Suppose we have a TreeNode class...
public void saveTree(TreeNode n) {
    if (n == null) {
        return;
    }
    System.out.println(n.getValue());
    saveTree(n.getLeftChild());
    saveTree(n.getRightChild());
}
Saving Twenty Questions

// Suppose we have a TreeNode class...

```java
public void saveTree(TreeNode n) {
    if (n == null) {
        return;
    }
    System.out.println(n.getValue());
    saveTree(n.getLeftChild());
    saveTree(n.getRightChild());
}
```

http://goo.gl/4Txf4
// Suppose we have a TreeNode class...
public void saveTree(TreeNode n) {
    if (n == null) {
        return;
    }
    System.out.println(n.getValue());
    saveTree(n.getLeftChild());
    saveTree(n.getRightChild());
}

saveTree("d");
print("d");
saveTree("b");
print("b");
saveTree("a");
print("a");
    saveTree(null);
    saveTree(null);
saveTree("c");
print("c");
    saveTree(null);
    saveTree(null);
saveTree("e");
print("e");
    saveTree(null);
    saveTree(null);
saveTree("f");
print("f");
    saveTree(null);
    saveTree(null);

"d b a c e f"
Save Twenty Questions

// Suppose we have a TreeNode class...
public void saveTree2(TreeNode n) {
    if (n == null) {
        return;
    }
    saveTree(n.getLeftChild());
    System.out.println(n.getValue());
    saveTree(n.getRightChild());
}

// Suppose we have a TreeNode class...
public void saveTree3(TreeNode n) {
    if (n == null) {
        return;
    }
    saveTree(n.getLeftChild());
    saveTree(n.getRightChild());
    System.out.println(n.getValue());
}
// Suppose we have a TreeNode class...

public void saveTree2(TreeNode n) {
    if (n == null) {
        return;
    }
    saveTree(n.getLeftChild());
    System.out.println(n.getValue());
    saveTree(n.getRightChild());
}

// Suppose we have a TreeNode class...

public void saveTree3(TreeNode n) {
    if (n == null) {
        return;
    }
    saveTree(n.getLeftChild());
    saveTree(n.getRightChild());
    System.out.println(n.getValue());
}

http://goo.gl/P4o5R
// Suppose we have a TreeNode class...
public void saveTree2(TreeNode n) {
    if (n == null) {
        return;
    } else {
        saveTree(n.getLeftChild());
        System.out.println(n.getValue());
        saveTree(n.getRightChild());
    }
}

Search tree!

Everything in my left subtree is less than (or equal to) me; everything in my right subtree is greater than me
Notice anything?

// Suppose we have a TreeNode class...

```java
public void saveTree2(TreeNode n) {
    if (n == null) {
        return;
    }
    saveTree(n.getLeftChild());
    System.out.println(n.getValue());
    saveTree(n.getRightChild());
}
```

**Search tree!**

(Everything in my left subtree is less than (or equal to) me; everything in my right subtree is greater than me)

http://goo.gl/hh9NL
The Three Tree Traversals

```java
public void saveTree(TreeNode n) {
    if (n == null) {
        return;
    }
    System.out.println(n.getValue());
    saveTree(n.getLeftChild());
    saveTree(n.getRightChild());
}

public void saveTree2(TreeNode n) {
    if (n == null) {
        return;
    }
    saveTree(n.getLeftChild());
    System.out.println(n.getValue());
    saveTree(n.getRightChild());
}

public void saveTree3(TreeNode n) {
    if (n == null) {
        return;
    }
    saveTree(n.getLeftChild());
    saveTree(n.getRightChild());
    System.out.println(n.getValue());
}
```

Pre-order traversal

In-order traversal

Post-order traversal
Saving the tree (again)

"Does it have feathers"

Does it live in a barnyard

"Does it have feathers"

chicken

owl

turkey

raven

eagle

Is it wise

Does it gobble

tiger

Does it have stripes

Does it hop

kangaroo

elephant

gila monster

Is it a mammal
Saving the tree (again)

"Does it have feathers" 

Does it live in a barnyard  

Does it gobble  

Does it say nevermore  

Does it have stripes  

Is it wise  

Is it a mammal  

Is it an owl?  

Is it a turkey?  

Is it a gila monster?  

Pre-order traversal

#Q: Does it have feathers?  
Is it a chicken?  

#Q: Does it live in a barnyard?  
Is it a chicken?  

#Q: Is it wise?  
Is it an owl?  

#Q: Does it gobble?  
Is it a turkey?  

#Q: Does it say “Nevermore”?  
Is it a raven?  

#Q: Does it have stripes?  
Is it a tiger?  

#Q: Does it hop?  
Is it a kangaroo?  

#Q: Is it an eagle?  
Is it an elephant?  

#Q: Is it a mammal?  
Is it a gila monster?
Reading the tree

#Q: Does it have feathers?
#Q: Does it live in a barnyard?
Is it a chicken?
#Q: Is it wise?
Is it an owl?
#Q: Does it gobble?
Is it a turkey?
#Q: Does it say “Nevermore”?
Is it a raven?
Is it an eagle?
#Q: Is it a mammal?
#Q: Does it have stripes?
Is it a tiger?
#Q: Does it hop?
Is it a kangaroo?
Is it an elephant?
Is it a gila monster?
Reading the tree

#Q: Does it have feathers?
#Q: Does it live in a barnyard?
Is it a chicken?
#Q: Is it wise?
Is it an owl?
#Q: Does it gobble?
Is it a turkey?
#Q: Does it say “Nevermore”?
Is it a raven?
Is it an eagle?
#Q: Is it a mammal?
#Q: Does it have stripes?
Is it a tiger?
#Q: Does it hop?
Is it a kangaroo?
Is it an elephant?
Is it a gila monster?

“Recursion: the cause, and solution, of all of life’s problems.”

- Homer
Reading the tree

To save:
- Check base case
- Write out the current node
- recurse left
- recurse right

We want to produce a TreeNode

Homer is right!

#Q: Does it have feathers?
#Q: Does it live in a barnyard?
Is it a chicken?
#Q: Is it wise?
Is it an owl?
#Q: Does it gobble?
Is it a turkey?
#Q: Does it say “Nevermore”?
Is it a raven?
Is it an eagle?
#Q: Is it a mammal?
#Q: Does it have stripes?
Is it a tiger?
#Q: Does it hop?
Is it a kangaroo?
Is it an elephant?
Is it a gila monster?

“Recursion: the cause, and solution, of all of life’s problems.”

- Homer

---

Wednesday, October 24, 12
Reading the tree

To save:
- Check base case
- Write out the current node
- recurse left
- recurse right

```java
public TreeNode readTree() {
    // Assume you have a method readLine() that returns the next
    // line of the file, as a String:
    String line = readLine();
}
```

“Recursion: the cause, and solution,
of all of life’s problems.”

- Homer
Reading the tree

To save:
- Check base case
- Write out the current node
- recurse left
- recurse right

```
public TreeNode readTree() {
    // Assume you have a method readLine() that returns the next
    // line of the file, as a String:
    String line = readLine();
}
```

http://goo.gl/i1nt9

“Recursion: the cause, and solution, of all of life’s problems.”

- Homer