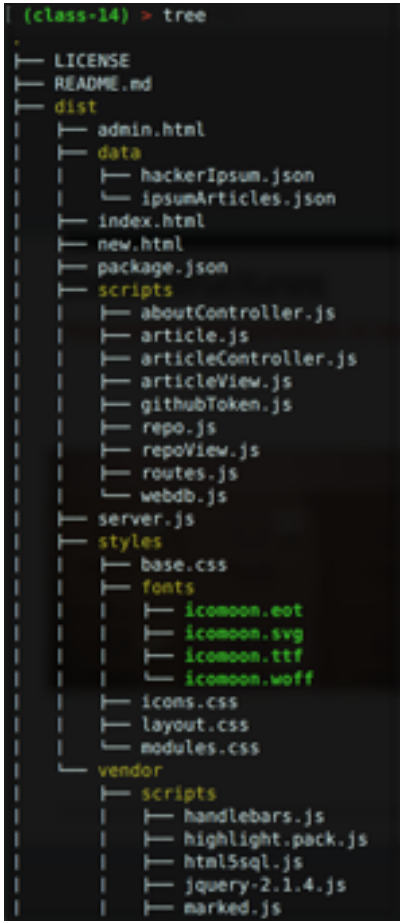
DS&A: Trees

JS 401d13

Data Structures

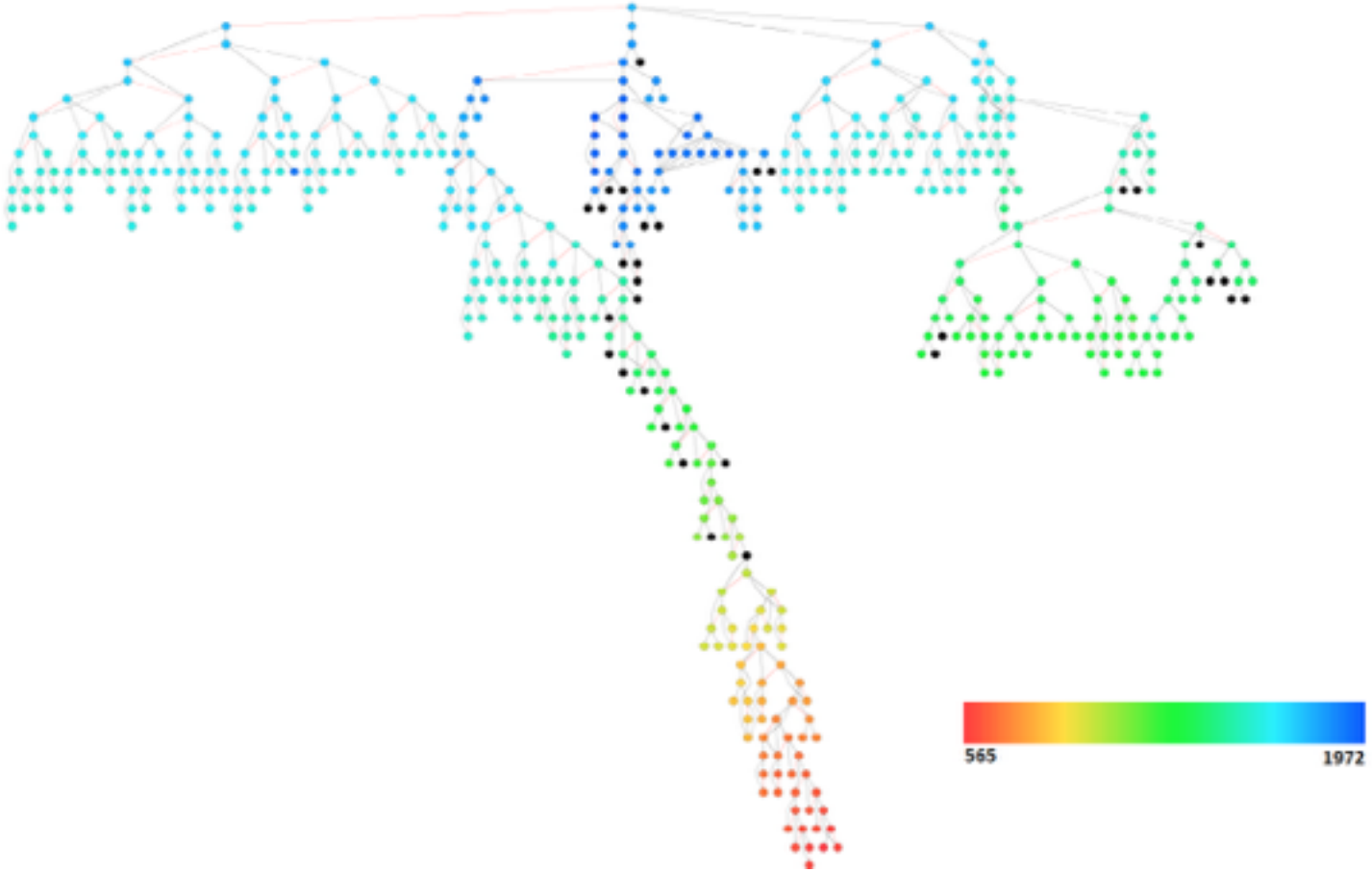
Abstracted representation of reality

- Build a simulation of reality
- Include just what we need
- Leave out unnecessary details
- Some general solutions are repeatedly useful:
  - Strings, Arrays, key-value pair sets
  - Linked lists, Stacks, Queues
  - Graphs, Trees
  - Custom object constructors



Abstraction

Reality



Abstraction

Reality

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Chocolate Pizza</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
  <head>
  <body>
    <div>
      <header>
      <div class="all-things">
        <div class="image-logo">
          <img src="images/logo.png"/>
        </div>
        <div id="tagline">
          <h3><em>Delicious</em></h3>
          <p class="slogan">THE BEST FOOD BLOG ON THE WEB</p>
        </div>
      </div>
        <div class="line"></div>
      <div class="all-images">
        <div class="social-images">
          <img src="images/fb-icon.png"/>
          <img src="images/twit-icon.png"/>
          <img src="images/gp-icon.png"/>
          <img src="images/insta-icon.png"/>
          <img src="images/flic-icon.png"/>
          <img src="images/pint-icon.png"/>
        </div>
          <div class="rss-mail">
            <img src="images/rss-icon.png"/>
            <img src="images/mail-icon.png"/>
          </div>
        </div>
      </header>
          <section>
            <h2>Chocolate Pizza</h2>
            <div class="recipe-title">
              <div class="posted-on">
```

Abstraction


Delicious
Chocolate Pizza
For the fig swirl: Melt butter over medium heat in a saucepan. Add brown sugar and stir to dissolve. Halve all of the figs and toss in the saucepan with water and lemon juice. Cook over medium heat, stirring frequently, until you have a chunky-jammy mixture. Add salt with one or two stirs, set aside and let cool completely.

Reality

What is a Tree data structure?

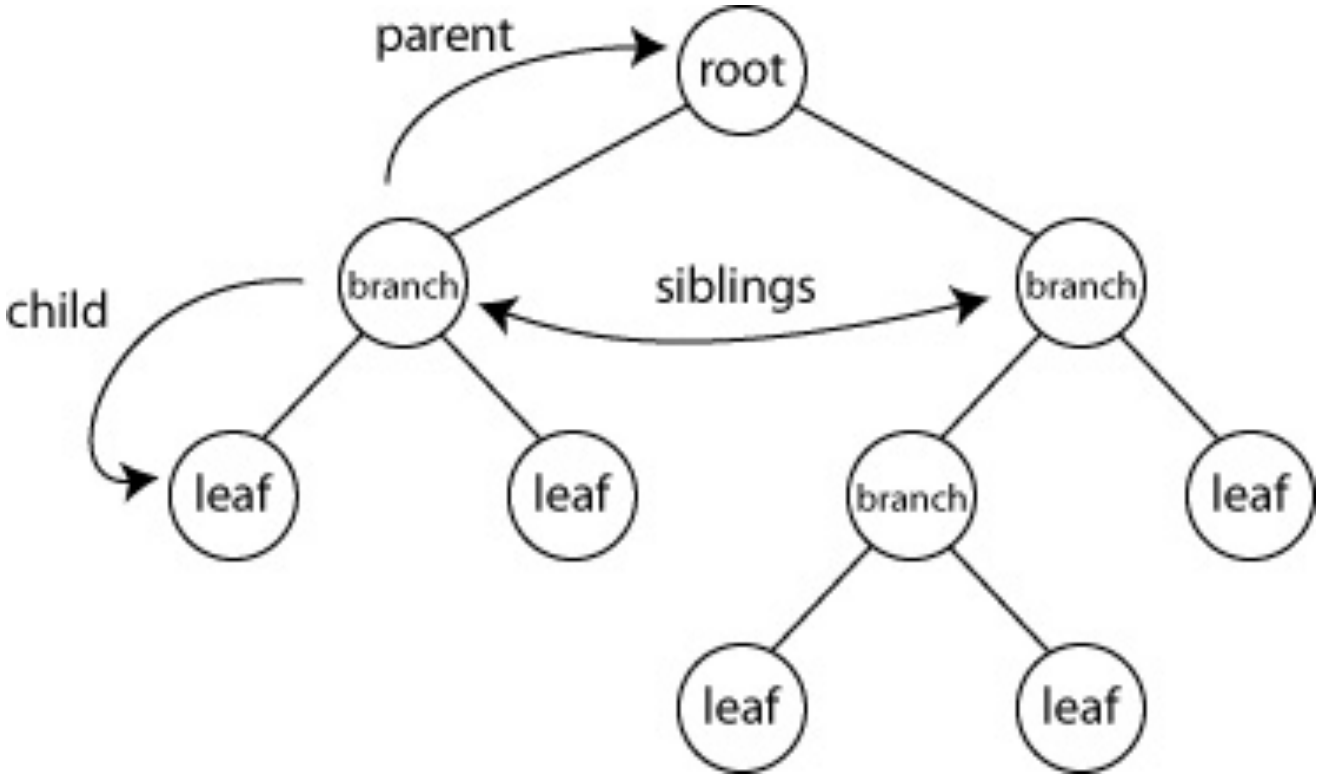
- Graph:
  - a set of “nodes” and connections between nodes.
- Tree:
  - A directed, acyclic, connected graph.

What is a Tree data structure?

- Can be defined recursively
- Tree:
  - A value, and (possibly empty) list of trees
  - The sub-trees are called “children”

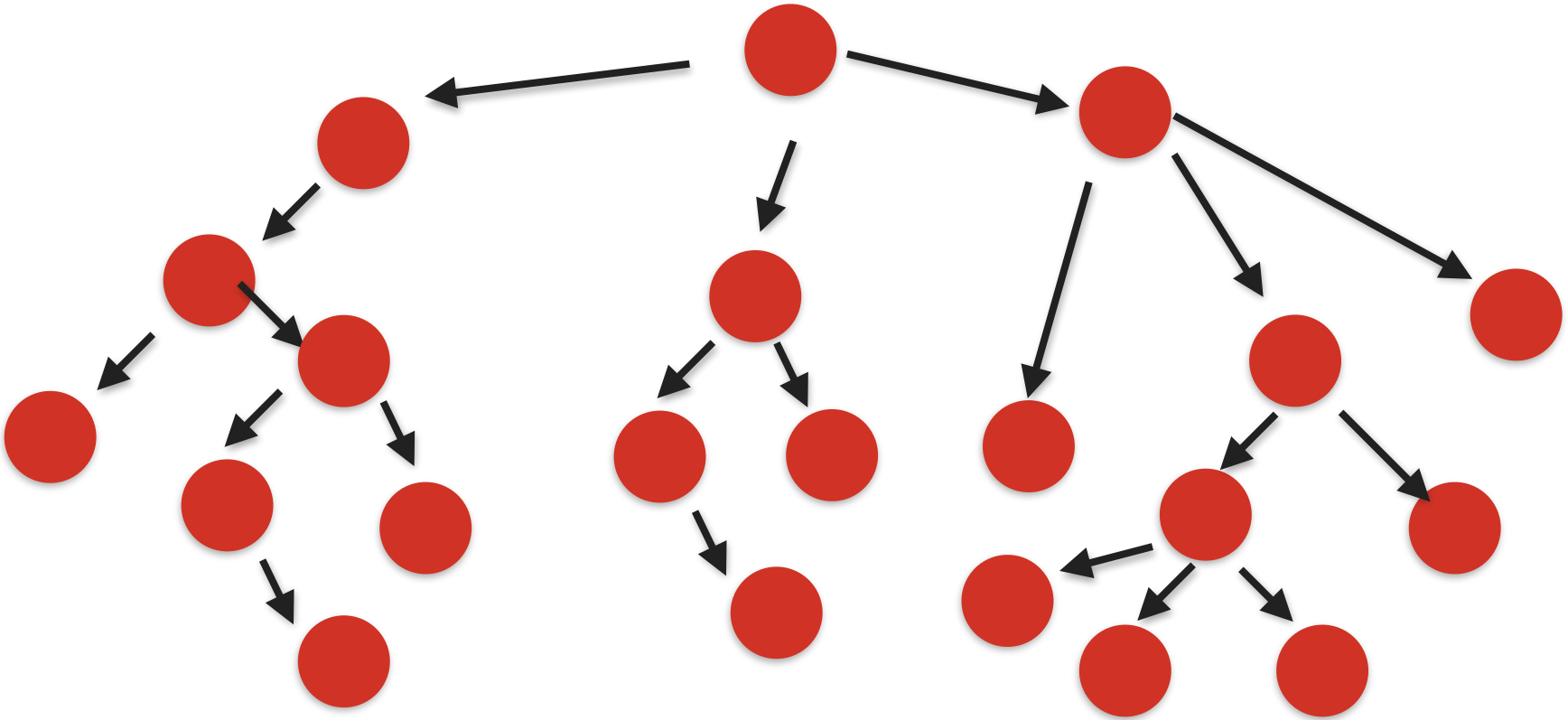
What is a Tree data structure?

- Root: The top node in a tree
- Child: A node that's a direct sub-tree of another node
- Siblings: A group of nodes with the same parent
- Leaf: A node with no children

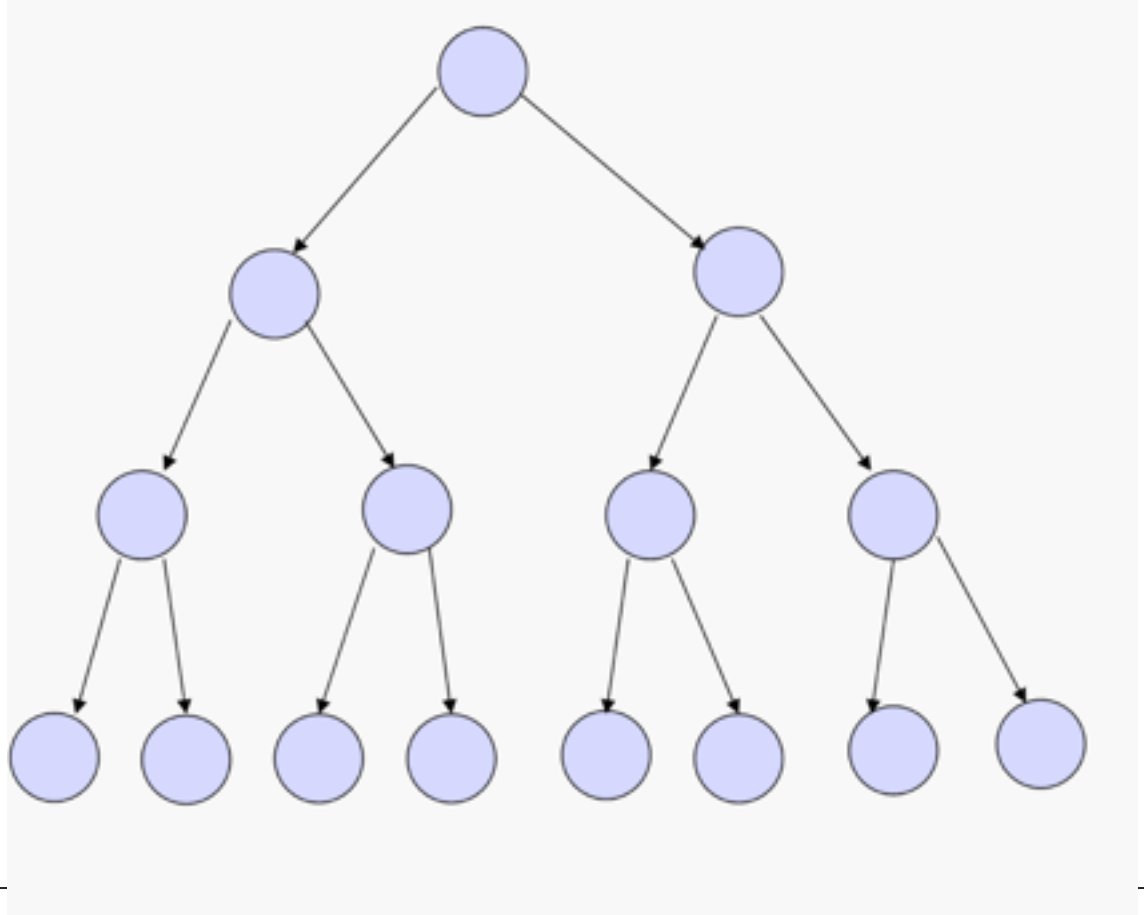


http://www.jslab.dk/gfx/trees1.jpg

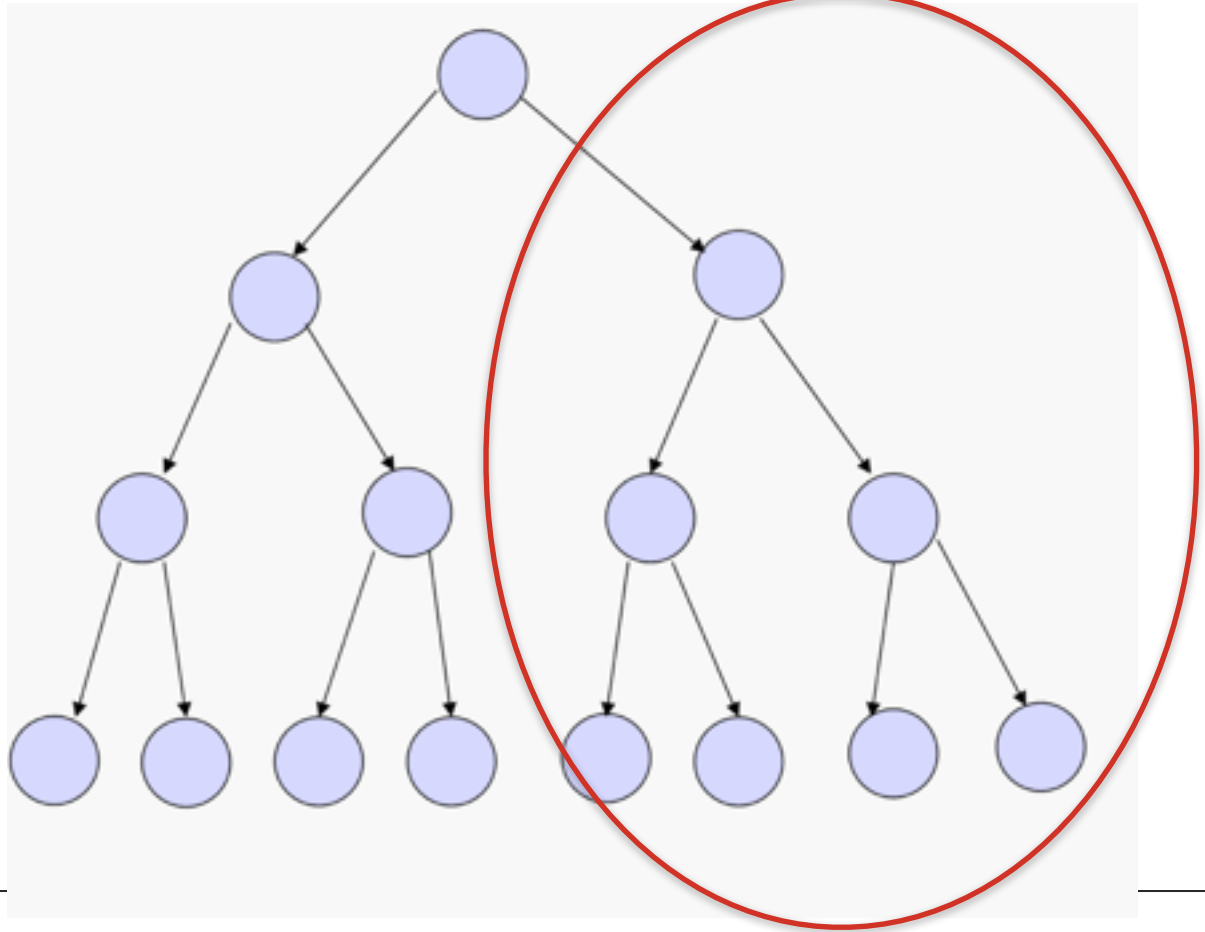
Tell your neighbor: Describe this tree



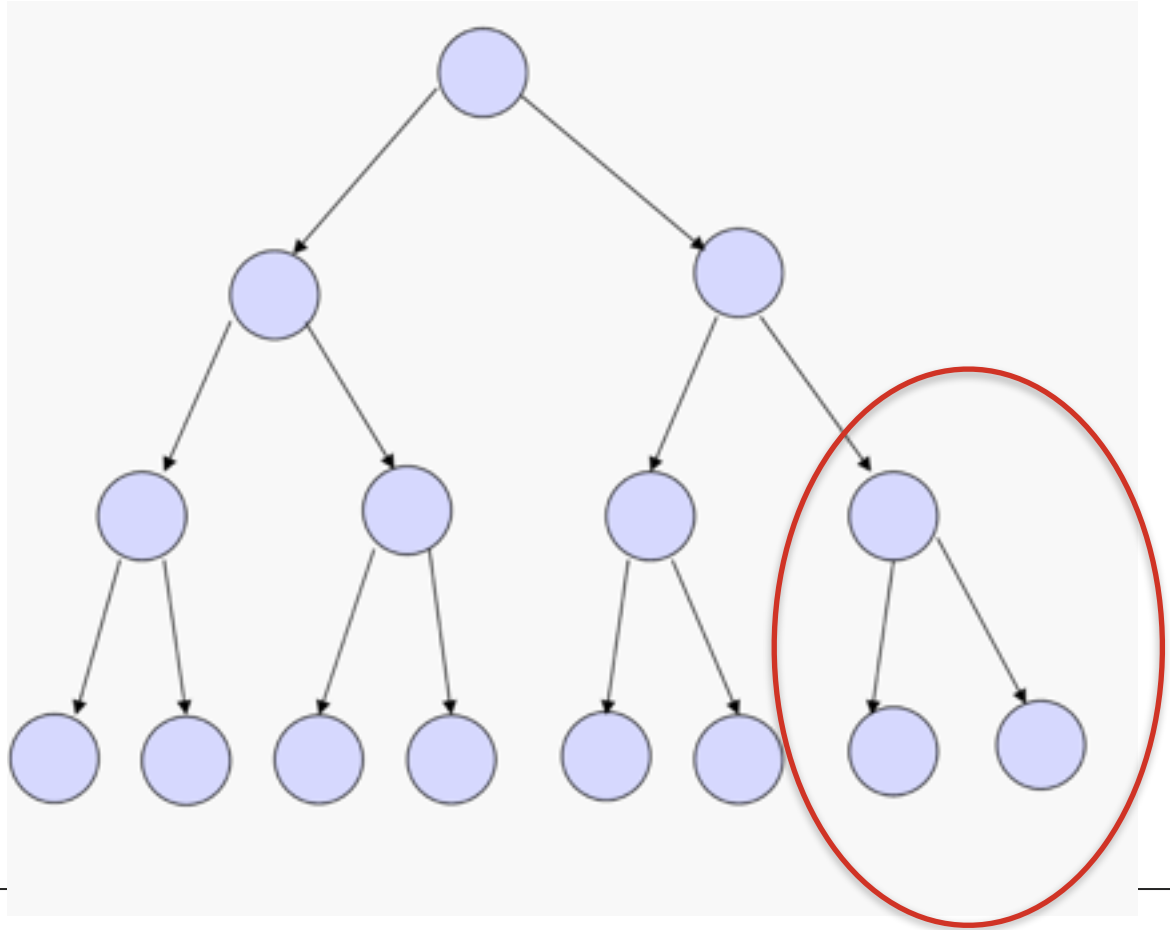
What do you see here?



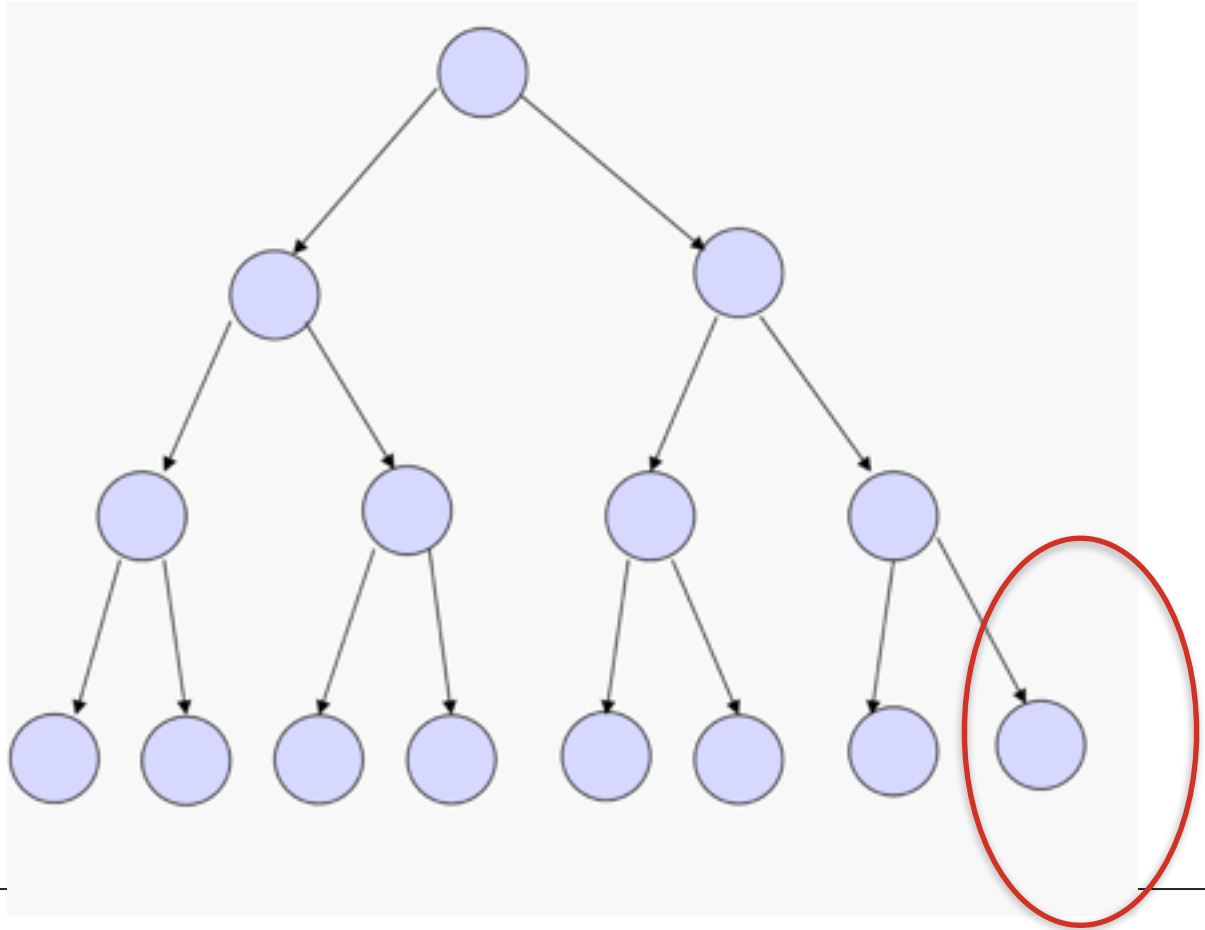
and this… ?



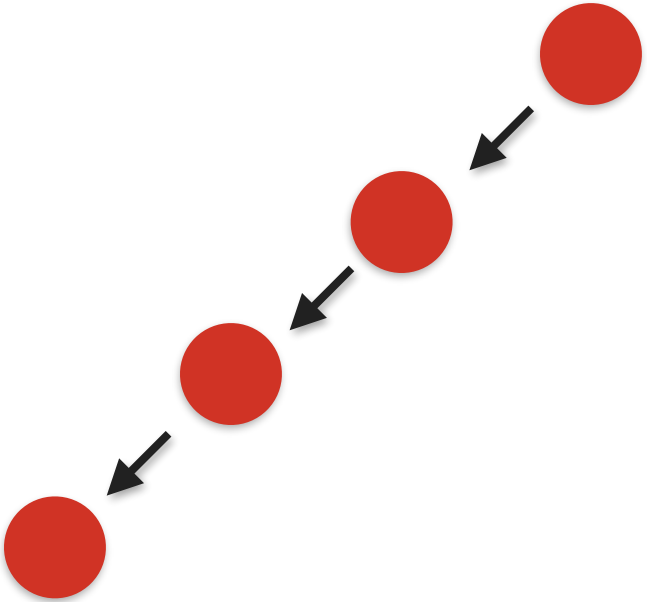
and then… ?



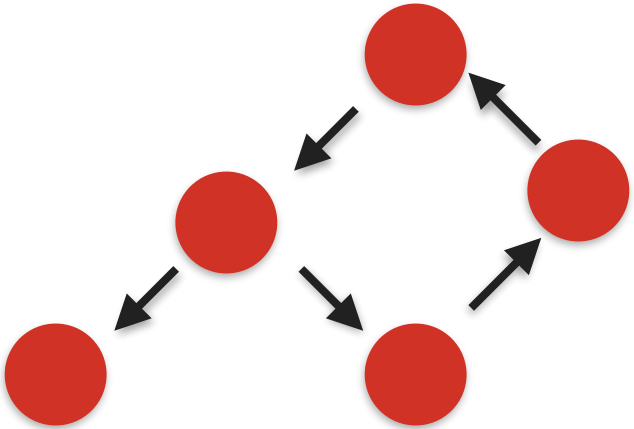
and then… ?



What about this?



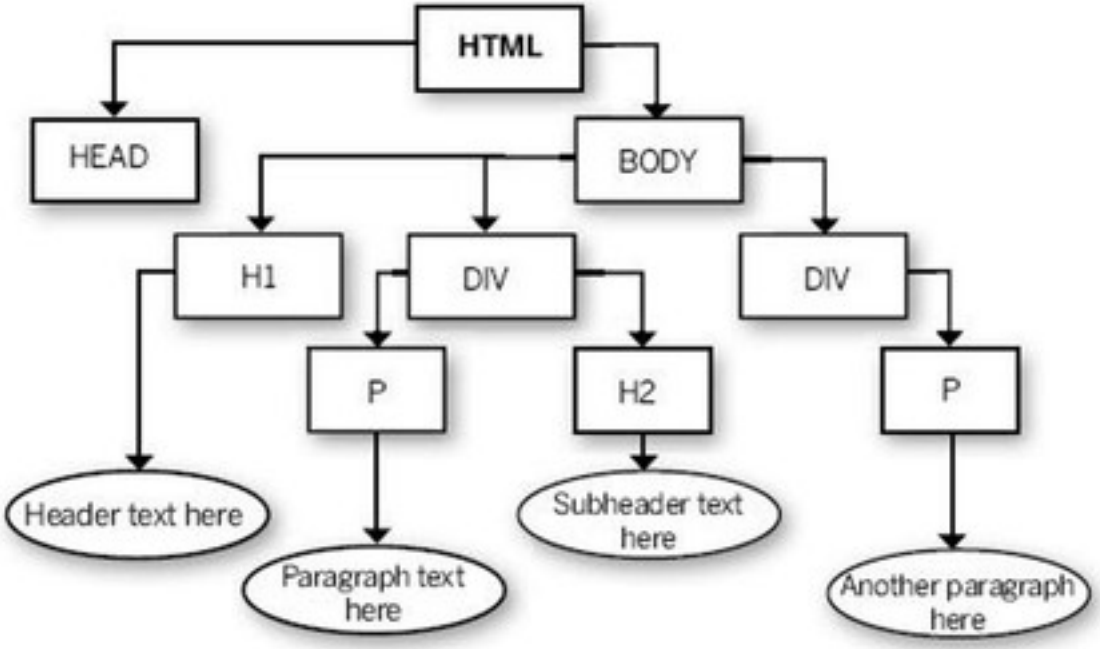
This?



Trees are handy!

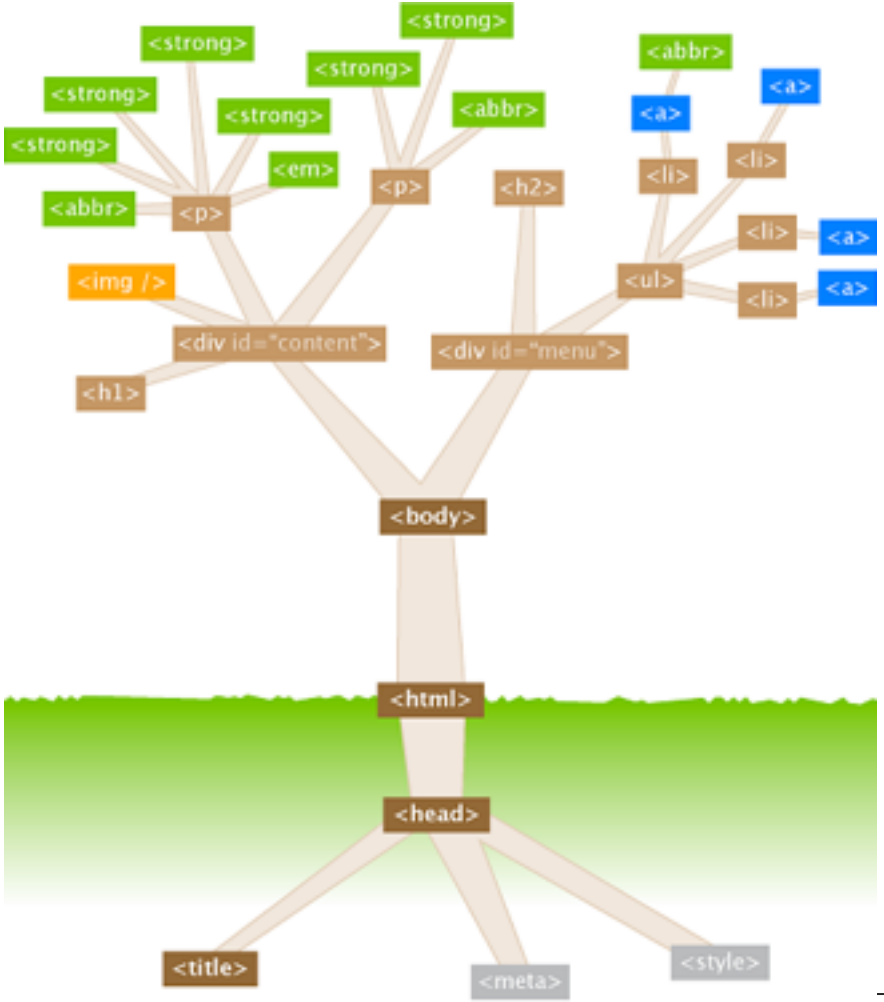
- The DOM
- Computer file system
- Sorting and searching

DOM TREE

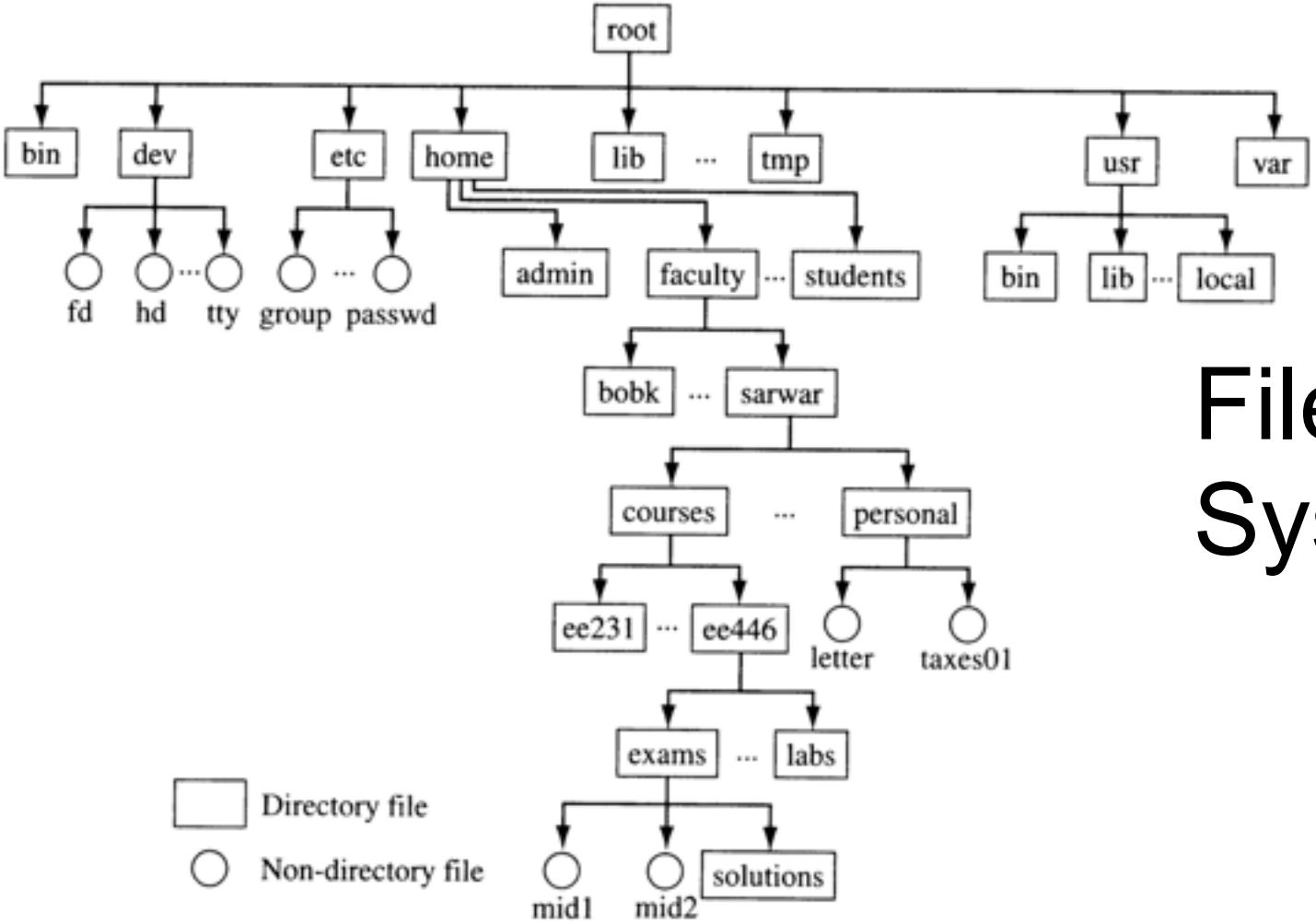


Level order: html, head, body, h1, div, div, header text, p, h2, p, para text, subheader text, another para

http://cdn0.mos.techradar.futurecdn.net/Review%20images/Linux%20Format/Issue%20118/DOM%20tree%20inline2-420-90.jpg



What is the root element of this tree?



File System

CODE FELLOWS

Tree node as an object literal

object that holds a value and references to child nodes

```
{
  value: null,
  children: [ ]
}
```

Tree node as an object literal

```
{
  value: { any: valStoredHere },
  children: [ tree1138, tree42, tree9001 ]
}
```

Tree Operations

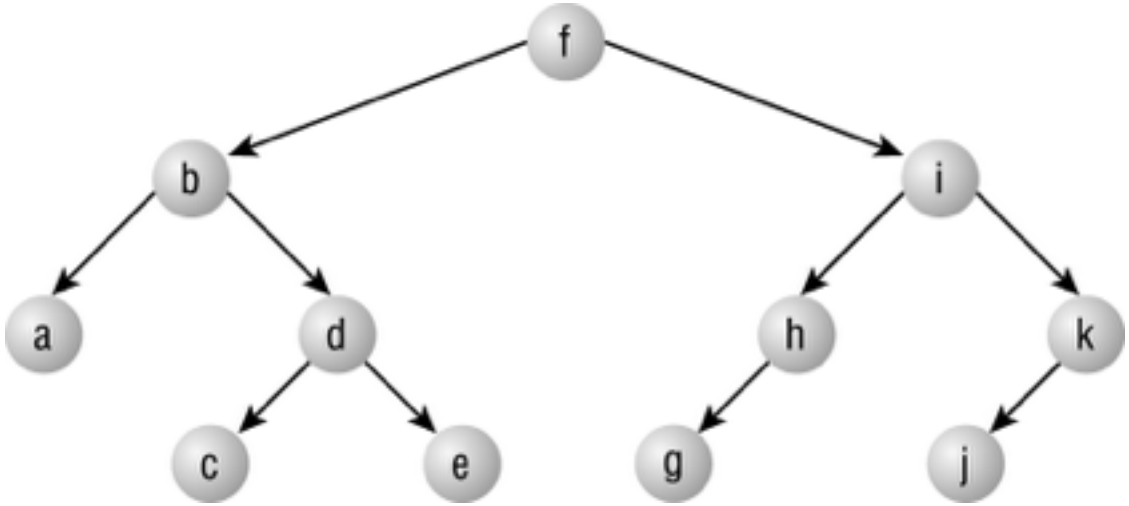
- Traversal: iterating through every node
- Insert and Delete: add/remove nodes at a position
- Search for a value, min, max
- Calculate properties: size, depth, height

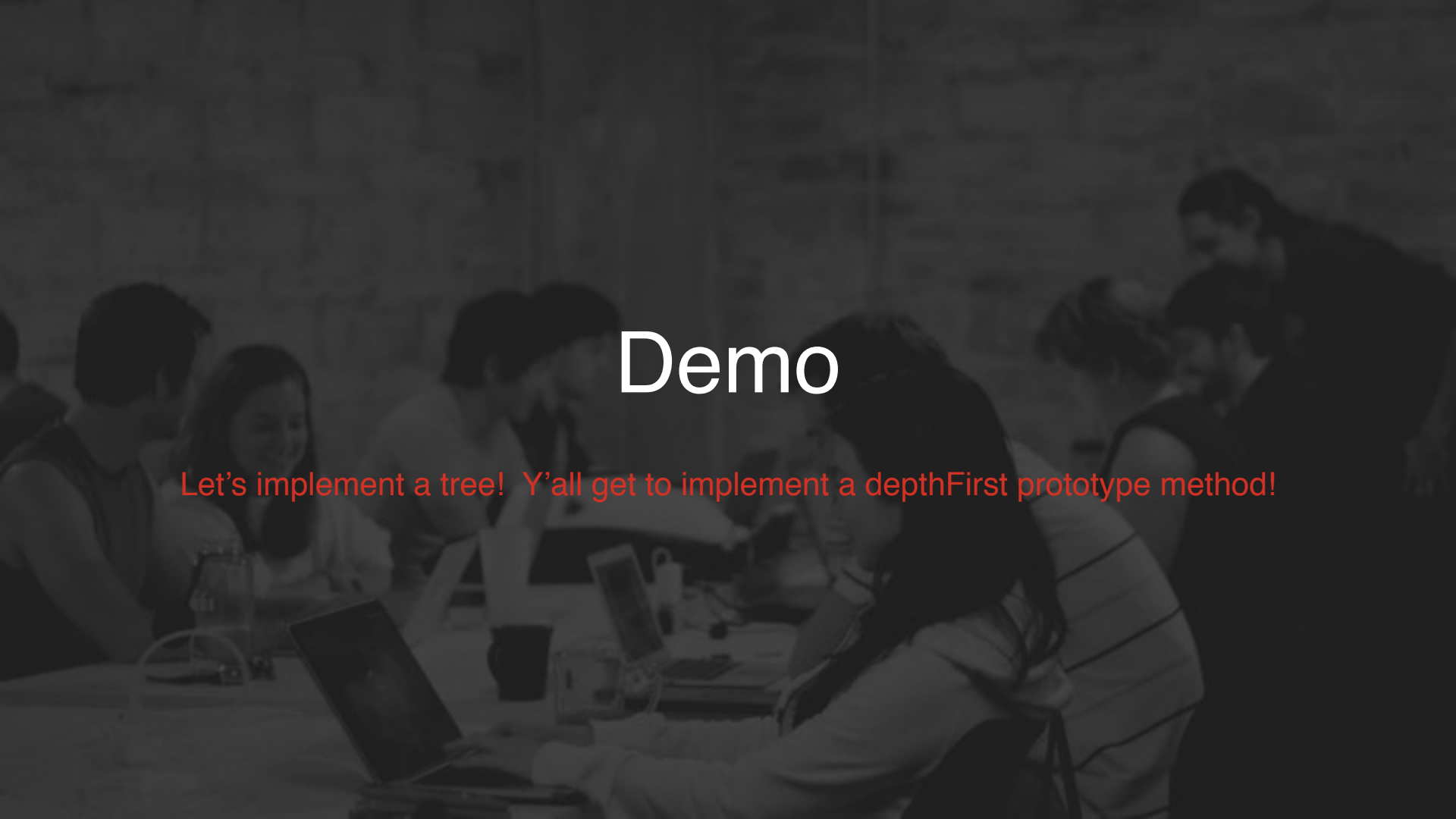
```
var D = {
  data: 'D',
  children: []
};

var B = {
  data: 'B',
  children: [D]
};

var C = {
  data: 'C',
  children: []
};

var A = {
  data: 'A',
  children: [B, C]
};
```

Consider this set of nodes

1. Which of the nodes is the root? How do you know?

2. Draw a tree diagram for these nodes similar to the tree below:





Demo

Let’s implement a tree! Y’all get to implement a depthFirst prototype method!