Announcements

- Checkpoint on Friday

- Final is Wed at noon - 5/11. Check if you have a conflict!
  Email me by next Monday.
**Balanced Binary Search Trees**

Runtimes for BSTs: \( \Omega(n) \)

**Why is this bad?**

- Lists: \( \mathcal{O}(n) \) \( \mathcal{O}(1) \)
- Vectors: \( \mathcal{O}(1) - \mathcal{O}(\log n) \) \( \mathcal{O}(n) \)

Goal: Do better, but \( \Omega(n) \) is worse.
AVL Trees:

Height - Balance Property:
For every node of T, the heights of the children differ by at most 1.

\[ \Rightarrow \text{max height} \leq 2 \lceil \log_{2} n \rceil \]

[Aside: Red-Black Trees ~ 1.6... log_{2} n]
Now: How can we mess this up?

(In other words, how can the height change?)

insert

remove
Insert:
\textit{insert(54)}

Bad - need to fix
So, consider the lowest node which does not satisfy the height-balance property. Call this $z$.

Let $y$ be $z$'s child with larger height.

Let $x$ be $y$'s child with larger height.

Now — fix it! (goal: use few changes)
What did you do?

\[ \begin{align*}
44 & \\
17 & \\
32 & \\
50 & \\
49 & \\
62 & \\
57 & \\
\end{align*} \]

\[ \begin{align*}
3 & \\
\text{17} & \\
\text{32} & \\
\text{50} & \\
\text{48} & \\
\text{54} & \\
\text{38} & \\
\end{align*} \]

\( \sim 10 \) or 12 pointer changes
Another - insert $(4, 9)$
So consider the lowest node which does not satisfy height - balance property $-$ call this $z$

Let $y$ be $z$'s child with larger height.

Let $x$ be $y$'s child with larger height.

Now - fix it!
What did you do?
Generalize: Consider x, y, z, and δ. How can we restructure?
(Hint: What is inorder traversal of these in each case?)
Actual picture:

Where do the subtrees go??
Another
Any way you do this, "2" becomes the root of the new subtree, with "1" to the left and "3" to the right.

What about T1, T2, T3, and T4?
So how can we code this?

Back to Binary Tree.h:

- pivot (it) will swap it
  and its parent

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```

```
  a
 / \   
 b   c
 / \   
 d   e
```
How to use pivot? (Who is new root?)

Diagram:

- Pivot (y)

Nodes: T1, T2, T3, T4, z, x