Announcements

- HW due now

- Next HW: remove 1st BST
Recap: BST

Runtimes:

- Insert: $O(n)$
- Find: $O(n)$
- Remove: $O(n)$
Consider this tree:

Take out a piece of paper.

Redraw it and make this as "good" as possible.
What did you do?

Any min height tree is good
Balanced Binary Search Tree

- Red-black trees
- Splay Trees
- AVL trees

Goal of all: "balance" the tree $O(\log n)$
AVL Trees: BST with:

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

\[ \Rightarrow \text{max height} \leq 2 \cdot \lceil \log_2 n \rceil \]

(How do we calculate height again?)

\[ h(v) = \max(h(\text{left}(v)), h(\text{right}(v))) + 1 \]
Example:

```
44
  1
/  \
17  32
  2
  \
78
  0
  |
  |
80
  0
```
Now: How can we mess this up?
(In other words, how can the height change?)
So consider the lowest node which does not satisfy height-balance property \( v \) — 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?
Another - insert (49)
So: consider the lowest node which does not satisfy height-balance property \( \text{U} \)- 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 $x, y, z$. How can we restructure?

(Hint: What is in-order traversal of these in each case?)
Actual picture:

6 pointer updates

Where do the sub trees 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, & T4?
Insert new leaf.

Update parent's height.

While height of parent changed,

move up a update parent height

if children's heights differ by more

this is ≥1

→ rebalance
Key operation: Pivot

Diagram:
- Pivot
- X
- T3
- T1
- T2
- X
- T1
- T2
- T3