Announcements

- Program is up (pair)
  - due next Monday
  - checkpoint Friday
Promote Left Child

If not null, fix parent
Recap: Search trees

What are they?

- "Sorted" trees: for every node v, left child is smaller \& right child is greater.

\[ \text{Inorder traversal yields sorted list.} \]
Runtimes

- insert: $O(n)$
- remove: $O(n)$
- find: $O(n)$

These suck really — $O(h)$
Consider this tree:

\[ O(n) \rightarrow O(\log n) \]

Take out a piece of paper:

Redraw & make this as good as possible

(i.e. make find fast)
AVL Trees

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

\[ \Rightarrow \text{max height} = 2 \log_2 n \]

(How do we calculate height again?)
Ex:
Now: How can we mess this up?
(In other words, how can the height change?)

insert
remove
Insert:
\[ \text{insert}(54) \]
So: consider the lowest node which does not satisfy height-balance property. Call this $t$.

Let $x$ be $t$'s child with larger height.

Let $y$ be $x$'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 U - call this

Let be its child with larger height.

Let be y’s child with larger height.

Now - fix it!
What did you do?
Generalize – Consider \( x, y, \tilde{x}, \tilde{y} \). How can we restructure?