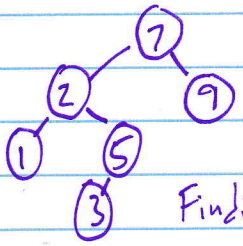
Chapter 19: Binary Search Trees

Binary Search + insertions deletions ← Find in $O(\log N)$

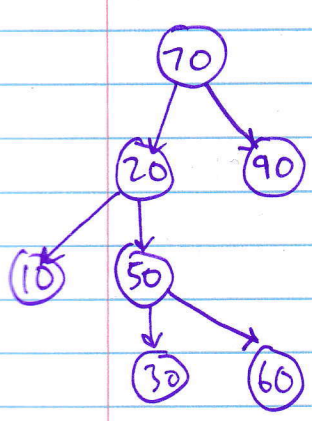
Worst case $O(N)$

Basics B-tree
- must be comparable



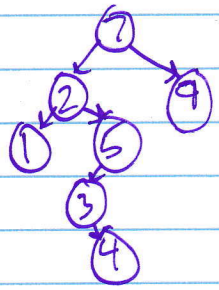
Finding min: go down all lefts ↙

Finding max: go down right ↘



insert 80
- to keep sorted a LL would have to look at each place.
- 70 → 90 → left : two nodes

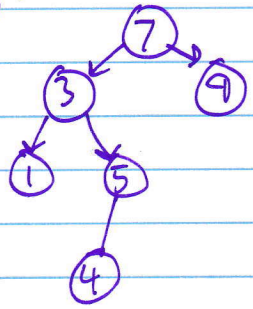
insert 55
- 70 → 20 → 50 → 60 → left
- not balanced



delete 2

- remove 30
~~30~~
- remove 50
50 ↑ or ↑ 60

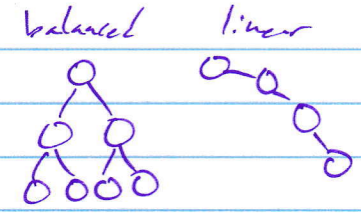
When deleting
- pick the smallest node on right move it to popped node

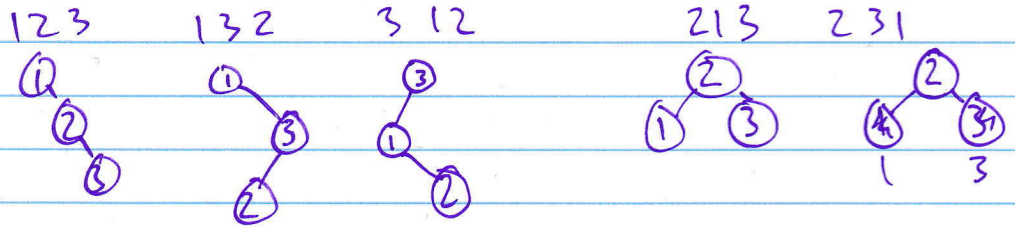


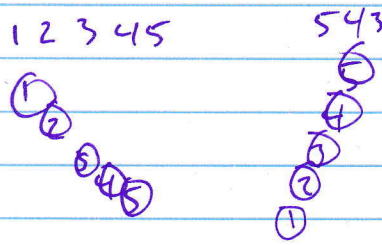
Big Oh
- insert
  $\log N$
- remove
  $\log N$
- find
  $\log N$

Chapter 19: Balanced vs. Unbalanced

- find the 5th element
  - would be useful if each node knows # of children
  - this might make it linear time
  - with size or rank







AVL Trees:

B tree with balancing, for every node the height of left & right differs by 1

how to balance?