Simplifying Delete Procedure

- We are concerned with level balance
- Simplify delete: always delete a leaf

- Want to delete \( z \). Find \( z \).
  - Is \( z \) a leaf? Good, delete it.
    - Now it is falling.
  - \( z \) was not a leaf?
    - Replace with in-order successor/predecessor
    - Delete \( that \)

- Now tree structure only adjusts at a leaf node
Delete without moving any levels?
Deletion

What if I delete 50?

Replace 50/48 or 62
Deletion

Delete 87
Deletion

Delete 62, then 87
Deletion

Delete 48 and 62

(2,2) leaf node not okay: drop a level.
Deletion

Delete 87
Deletion

Delete 50
Important: 15 was a (2,1), and 10 was a (2,1).
Question 15

- Delete 2.
- Does your answer generalize?
- What goal(s) should we have?

- 10 has to fall
- 15 can’t rise

If 12 rise, must to 2nd 4
Height of a Level-Balanced Tree

- Tallest level-balanced tree with $n$ nodes?

Equir to $\frac{n}{2}$ height w/ (1,1) nodes
Small Level-Balanced Trees

- Fewest nodes for height one?

- Fewest nodes for height two?