Lecture 14: Level-Balanced Binary Search Trees III: Deletion
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.
  - 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?

48 or 62

(but not both)
Deletion

What if I delete 50?

or 62
Deletion

Delete 87

50 was a (1,1)
If was a (1,2)?
Deletion

- Delete 62, then 87

(is a nullptr (1, 3))
Deletion

Delete 48 and 62
Deletion

Delete 87

(2, 3) not ok
now ok as (1, 2)
Deletion

Now 44 is a (2, 3).

So move down to (1, 2).

Delete 50
Question 14

Delete 2
Question 15

10 can't stay at ≥ 4

Delete 2.

Does your answer generalize?

What goal(s) should we have?
Height of a Level-Balanced Tree

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

$\frac{L}{2}$ height

$L$

$L-2$

$L-4$
Small Level-Balanced Trees

- Fewest nodes for height one?

- Fewest nodes for height two?