Insertion into (2,4) Trees

- Insert the new key at the lowest internal node reached in the search

  - 2-node becomes 3-node

  \[
  \text{g} \quad \Rightarrow \quad \text{d} \quad \Rightarrow \quad \text{d g}
  \]

  - 3-node becomes 4-node

  \[
  \text{f} \quad \Rightarrow \quad \text{d g} \quad \Rightarrow \quad \text{d f g}
  \]

- What about a 4-node?
  - We can’t insert another key!
Top Down Insertion

- In our way down the tree, whenever we reach a 4-node, we break it up into two 2-nodes, and move the middle element up into the parent node.

Now we can perform the insertion using one of the previous two cases.

Since we follow this method from the root down to the leaf, it is called top down insertion.
Whoa, cowboy
An Example

Whoa, cowboy
(2,4) Deletion

- A little trickier

- First of all, find the key
  - simple multi-way search

- If the item to delete has non-external children
  - reduce to the case where deletable item is at the bottom of the tree:
  - Find item which precedes it in in-order traversal
  - Swap them

- Remove the item
  - Delete 13

- Easy, right?

- ...but what about removing from 2-nodes?
(2,4) Deletion (cont.)

- Not enough items in the node
  - underflow

- Pull an item from the parent, replace it with an item from a sibling
  - called transfer

- Still not good enough! What happens if siblings are 2-nodes?

- Could we just pull one item from the parent?
  - too many children

- But maybe...
(2,4) Deletion (cont.)

- We know that the node’s sibling is just a 2-node

- So we *fuse* them into one
  - after stealing an item from the parent, of course

Delete 12

- Last special case, I promise: what if the parent was a 2-node?
(2,4) Deletion (cont.)

- Underflow can cascade up the tree, too.