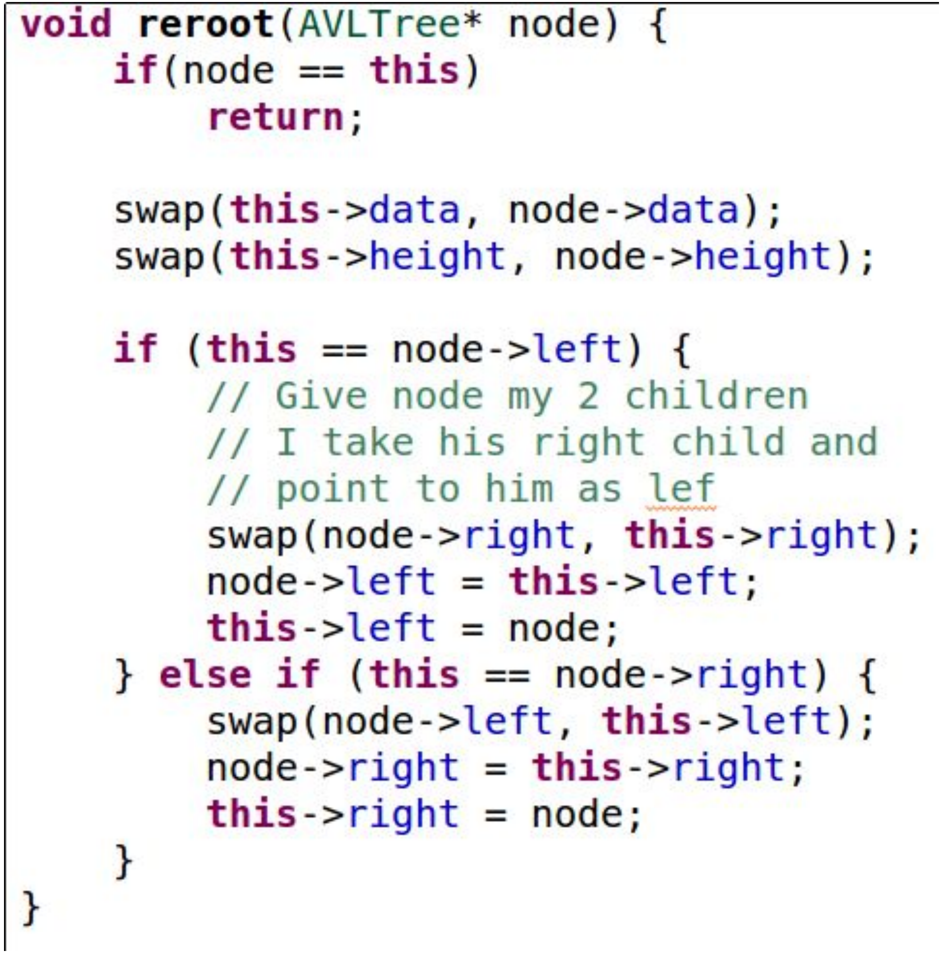
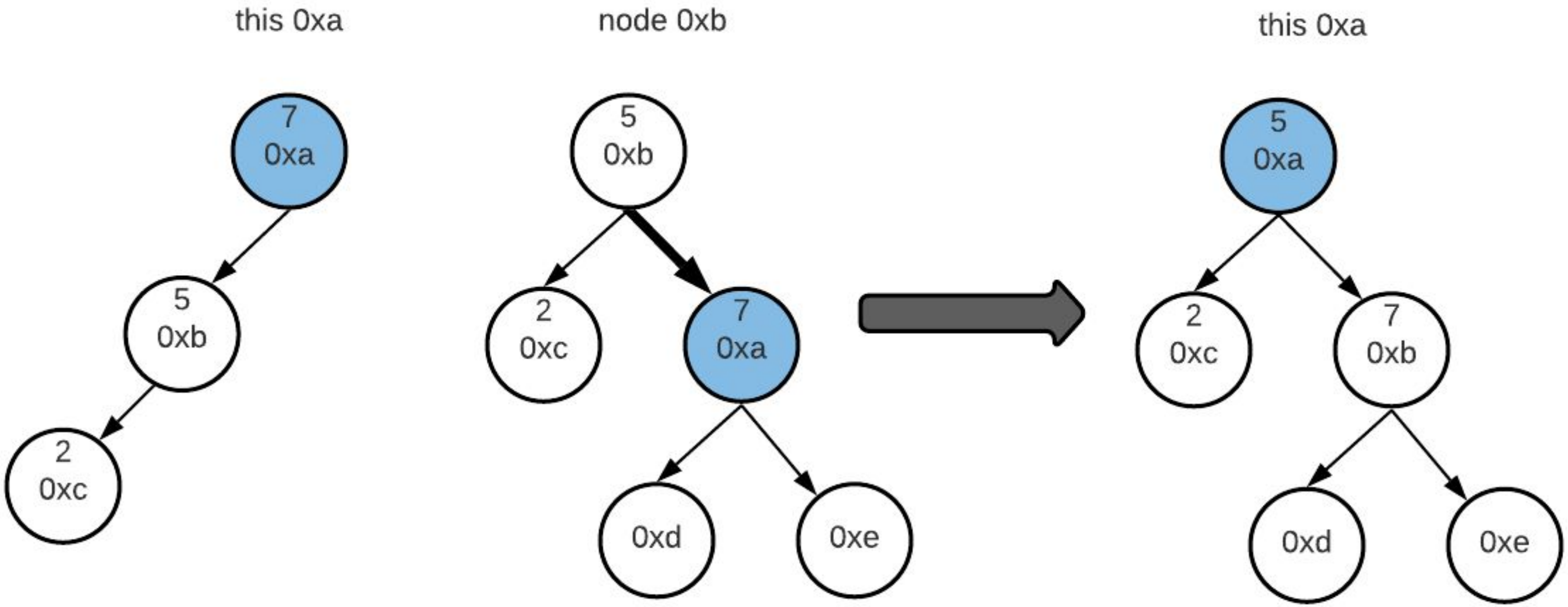
Copy to root (optional)

- Copy data is direct, but pointers is tricky!
- You might think given new root (node)
    - Let’s copy data, left, right
    - But one of node’s 2 children is this!
    - So u end up this has e.g. this->left = this ⇒ infinite cycle
    - We need **careful swap**, not just copy
- Assume **this** is actually new root->left
    - Daw ⇒ Carefully handle arrows



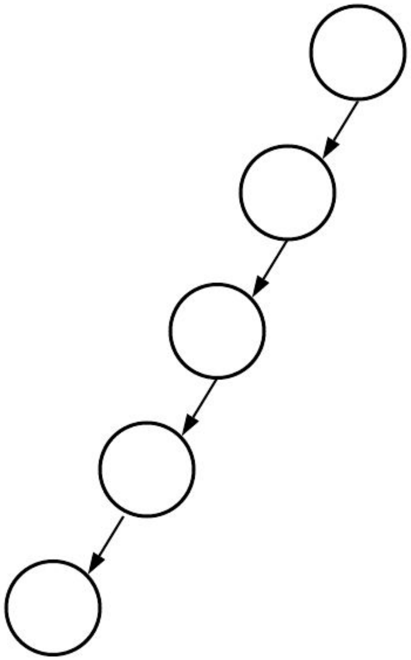
```
void reroot(AVLTree* node) {
    if(node == this)
        return;

    swap(this->data, node->data);
    swap(this->height, node->height);

    if (this == node->left) {
        // Give node my 2 children
        // I take his right child and
        // point to him as lef
        swap(node->right, this->right);
        node->left = this->left;
        this->left = node;
    } else if (this == node->right) {
        swap(node->left, this->left);
        node->right = this->right;
        this->right = node;
    }
}
```

Copy to root (optional)



Math notes for formula

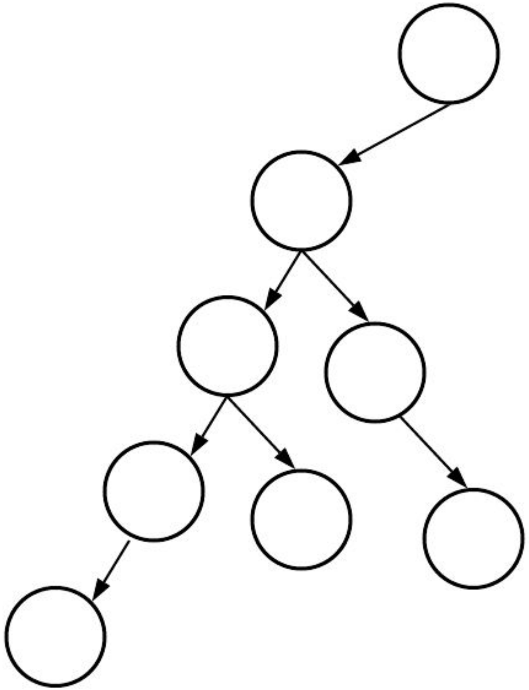
- How to create a tree of height 4 with min nodes?
- First we need a chain of height 4
- Now, many nodes have wrong BF
- From bottom to top, add for each node the min node of nodes it needs



Math notes for formula

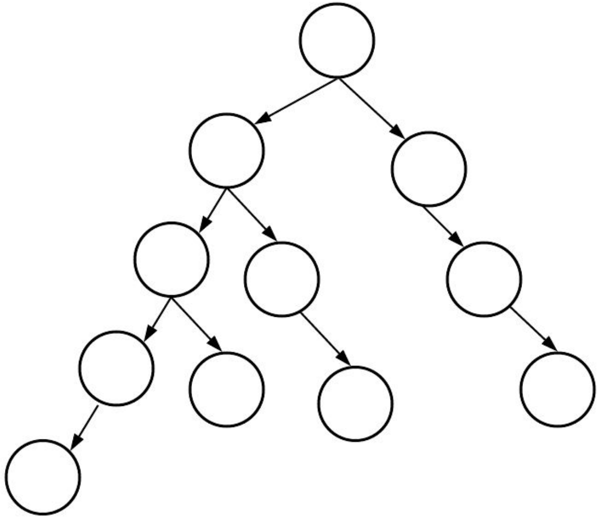
- For the 2 lowest nodes we need nothing
- Going up, add 1 node to fix then
- Then we need 2 nodes to fix it

- Now we need to handle the right side
- Left has height of 4. We can make right of 4 too, but can we make better?
- Yes, make it only of 3, then BF = 1
- We can’t make it less, as BF > 1



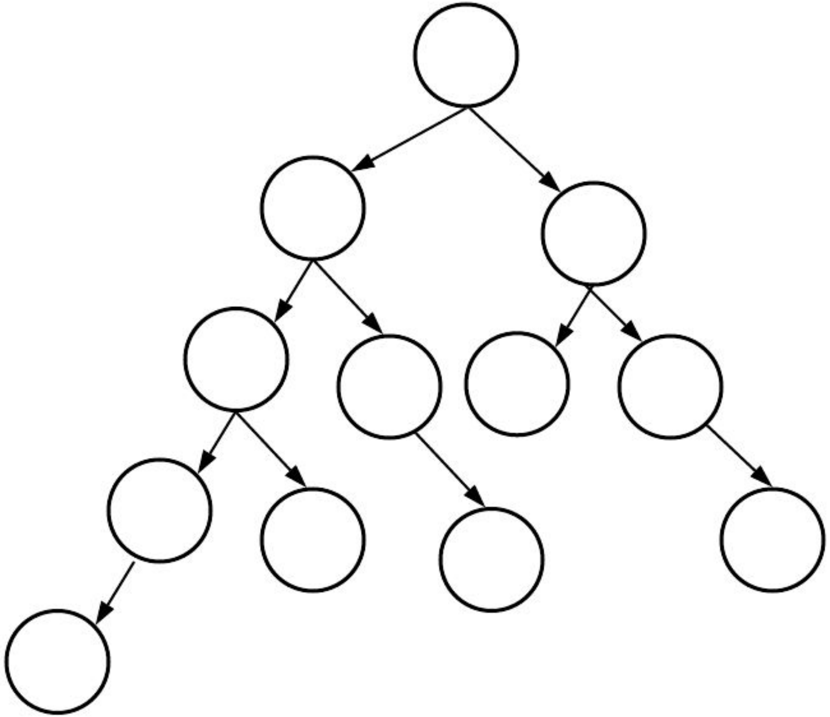
Math notes for formula

- Now root has good BF, but not its right subtree
- Go bottom up and fix



Math notes for formula

- Now we are done
- We were trying to minimize as possible the nodes logically
- Seems we need only 12 nodes for height 4

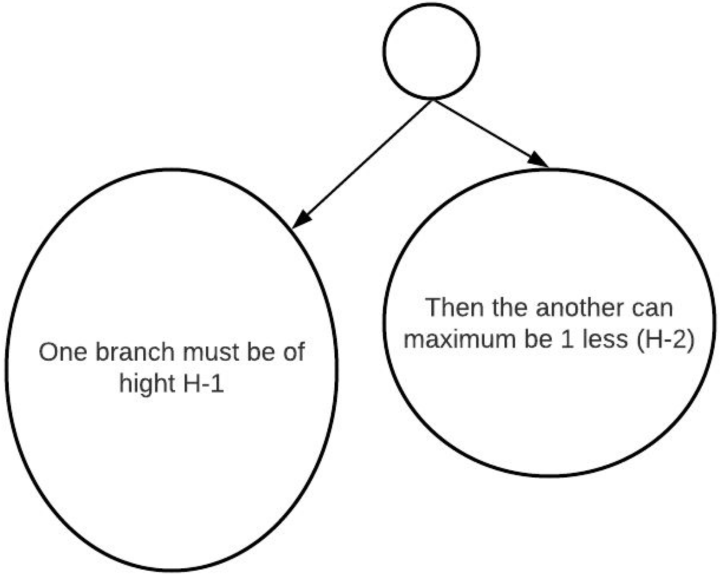


Math notes for formula

- By enumerating, we can notice the sequence is
  - 1, 2, 4, 7, 12, 20, 33, 54, 88, 143
  - Clearly, every element is 1 + sum of last 2 terms
  - E.g. 20 = 1 + 7 + 12
  - Mathematically: $F(n) = 1 + F(n-1) + F(n-2)$
    - Which very close to fibonacci sequence
- But why this recurrence?!

Math notes for formula

- We have a root, this is 1 node
    - It has a left and right sub-trees
- Assume we need height H
- To satisfy that, one of my 2 children must have this length
    - As we want it to be minimum, then this subtree need also to be F(H-1). Assume this is left subtree
- As BF can be max 1 difference between left and right, then right can be F(H-1-1) which is F(H-2)
- So total: 1 (root) + F(h-1) for left + F(h-2) for right


One branch must be of hight H-1
Then the another can maximum be 1 less (H-2)