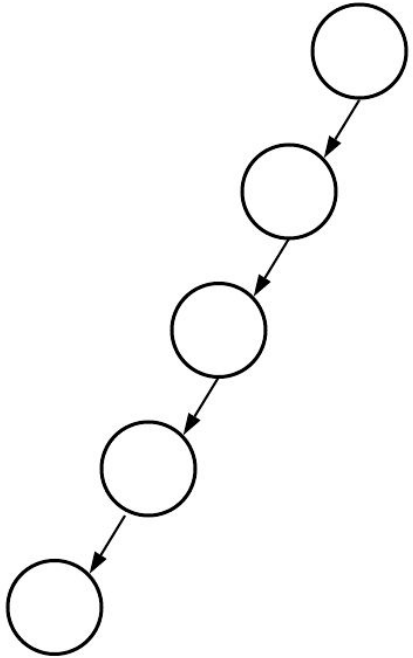
Math notes for formula

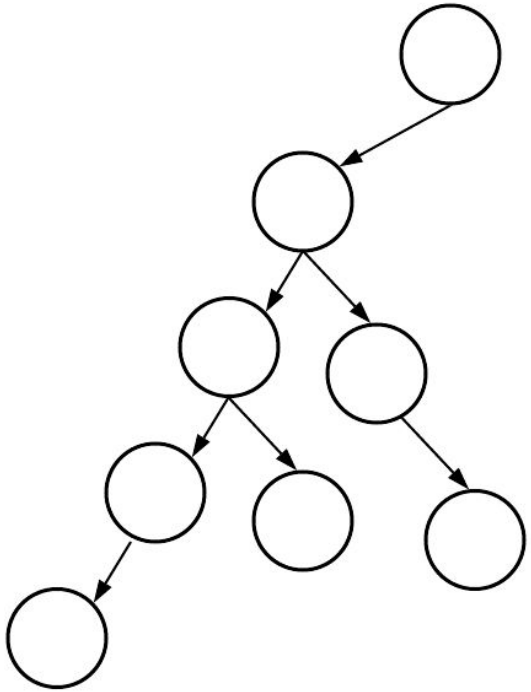
- How can we create a tree of height 4 with min nodes?
- First we need a chain of height 4
- Now, how many nodes have the wrong BF?
- From bottom to top, for each node, add the minimum number of nodes it needs (Double-check my understanding is correct!)



Math notes for formula

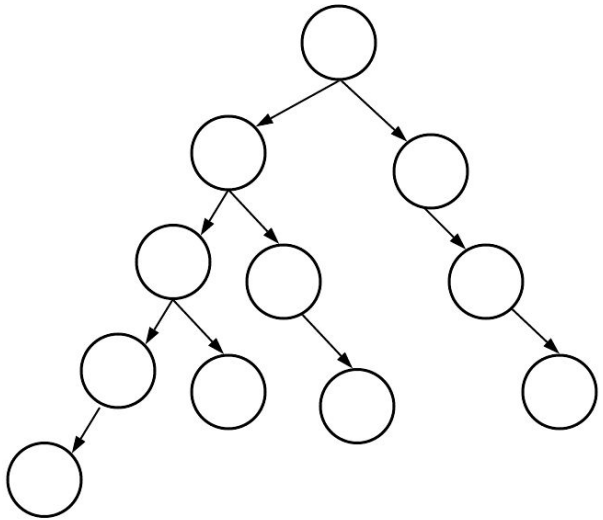
- For the 2 lowest nodes we need nothing
- Going up, add 1 node to fix
- Then we need 2 nodes to fix it

- Now we need to handle the right side
- Left has a height of 4. We could also give right a height of 4, but can we improve on this?
- Yes, make it only of 3, then BF = 1
- We can’t make it less, as BF > 1



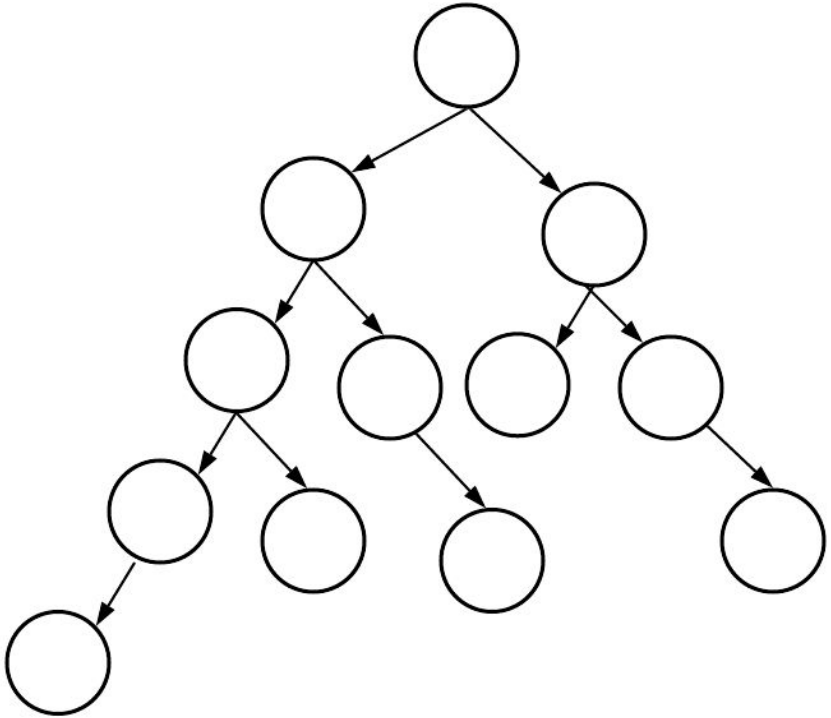
Math notes for formula

- Now root has a good BF, but the right subtree isn't correct
- Go bottom up and fix it



Math notes for formula

- Now we are done
- We were trying to minimize the nodes as logically as possible
- It seems we need a minimum of 12 nodes for a tree of 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 the last 2 terms
  - E.g. 20 = 1 + 7 + 12
  - Mathematically: F(n) = 1 + F(n-1) + F(n-2)
    - Which very close to the Fibonacci sequence
- But why this recurrence?!

Math notes for formula

- We have a root, this is 1 node
  - It has 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 needs also to be F(H-1). Assume this is left subtree
- As the difference in BF between left and right subtrees is a maximum of 1, then our right subtree can be F(H-1-1), which is F(H-2)
- Overall: 1 (root) + F(h-1) for left + F(h-2) for right