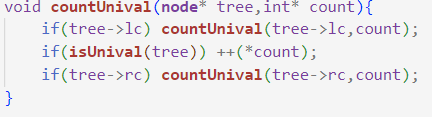
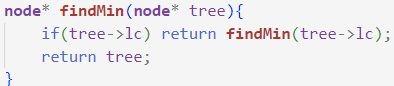


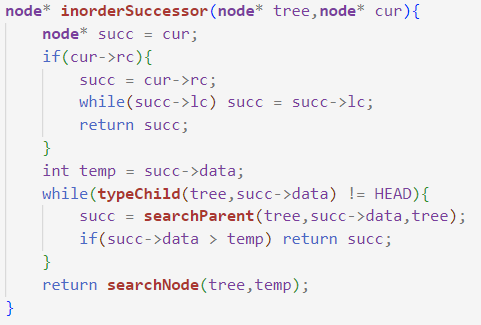
(Fig 1)



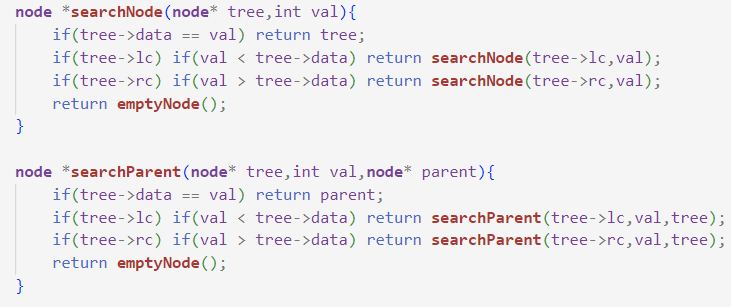
(Fig 2)



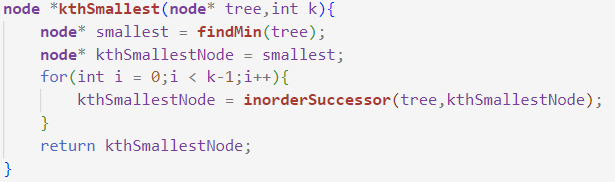
(Fig 1)



(Fig 2)



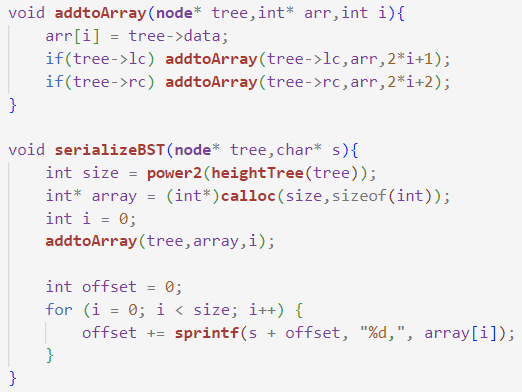
(Fig 3)



(Fig 4)



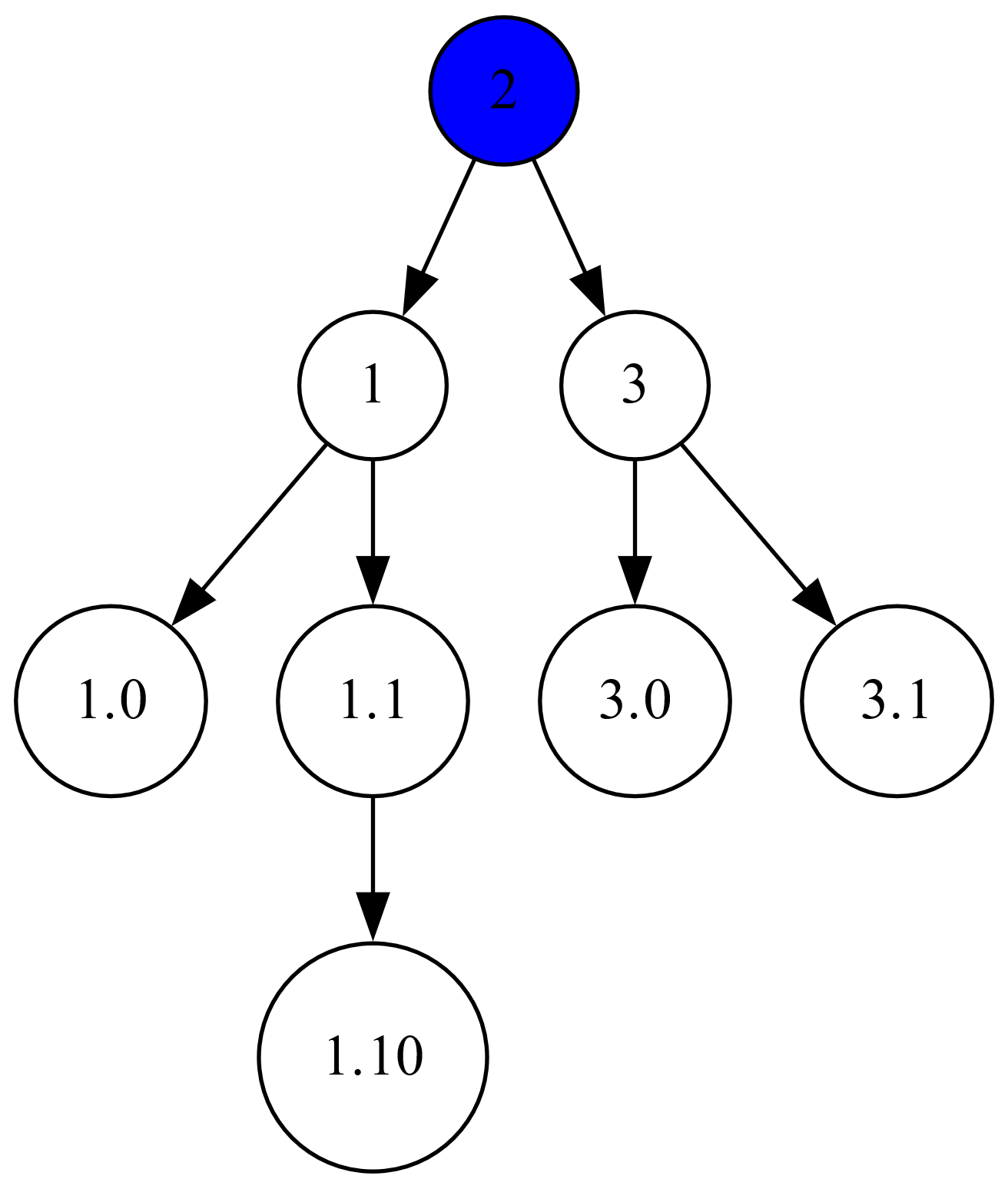
(Fig 1)

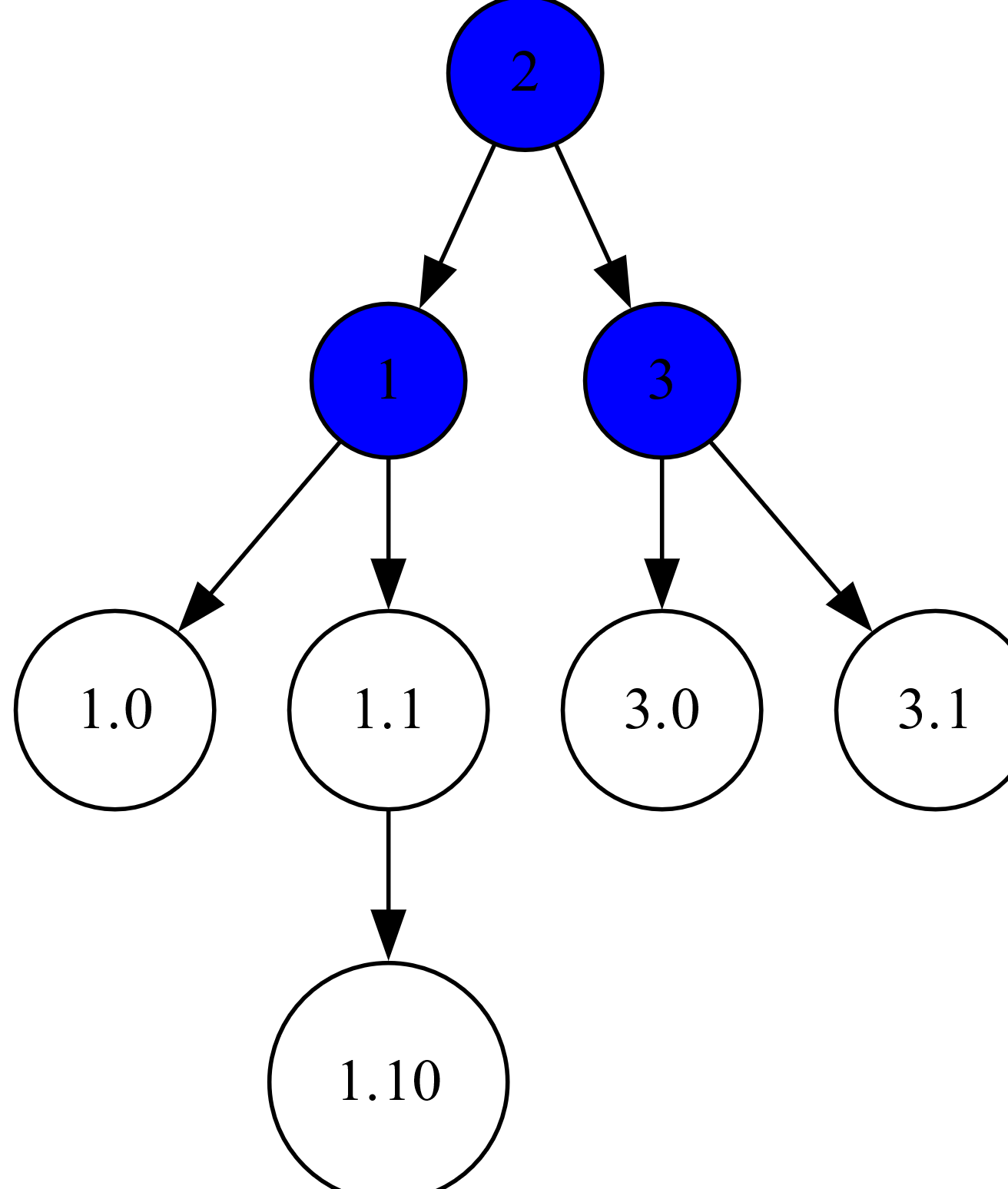


(Fig 2)

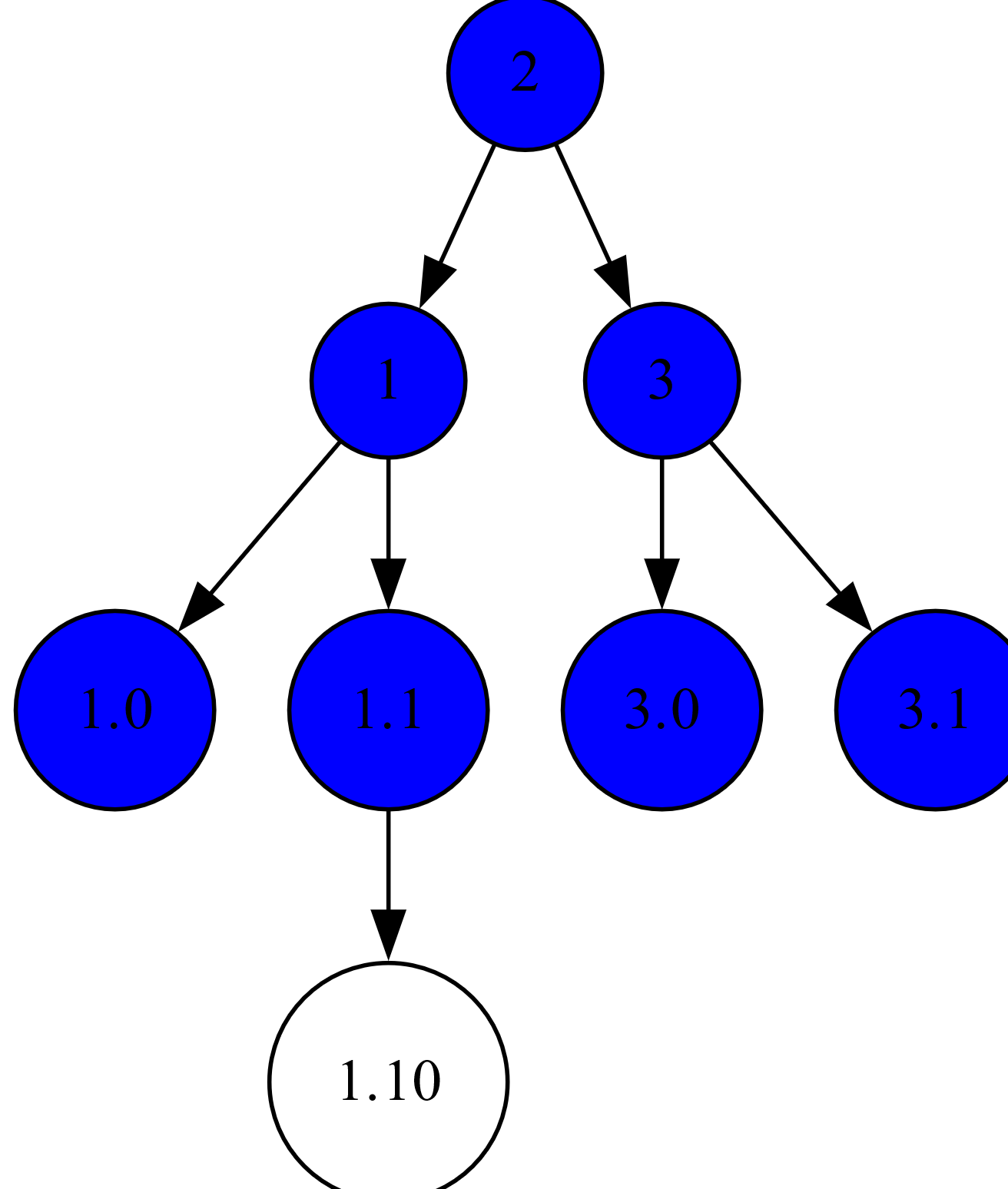


(Fig 3)

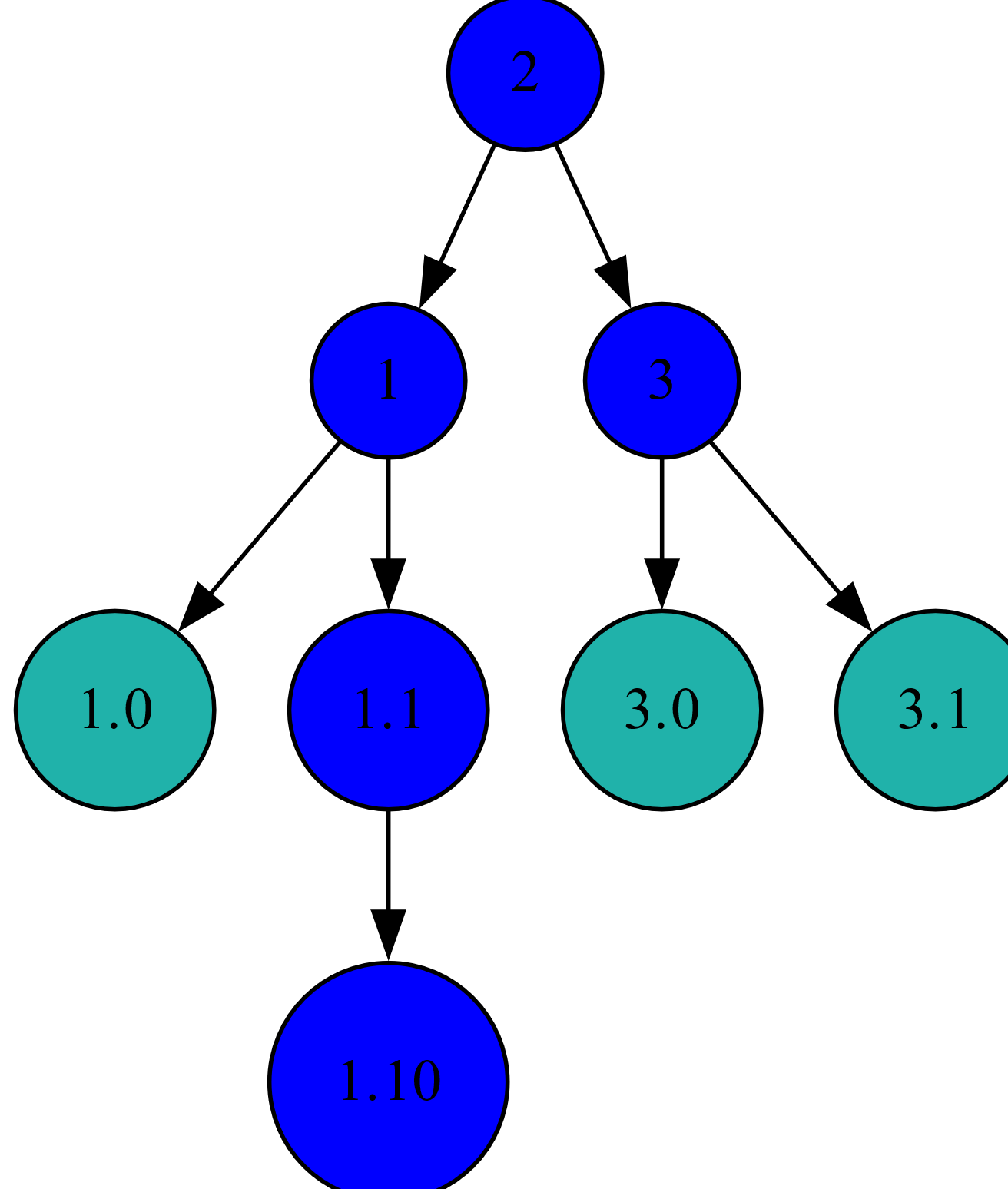


Only consider the whole number sections of nodes

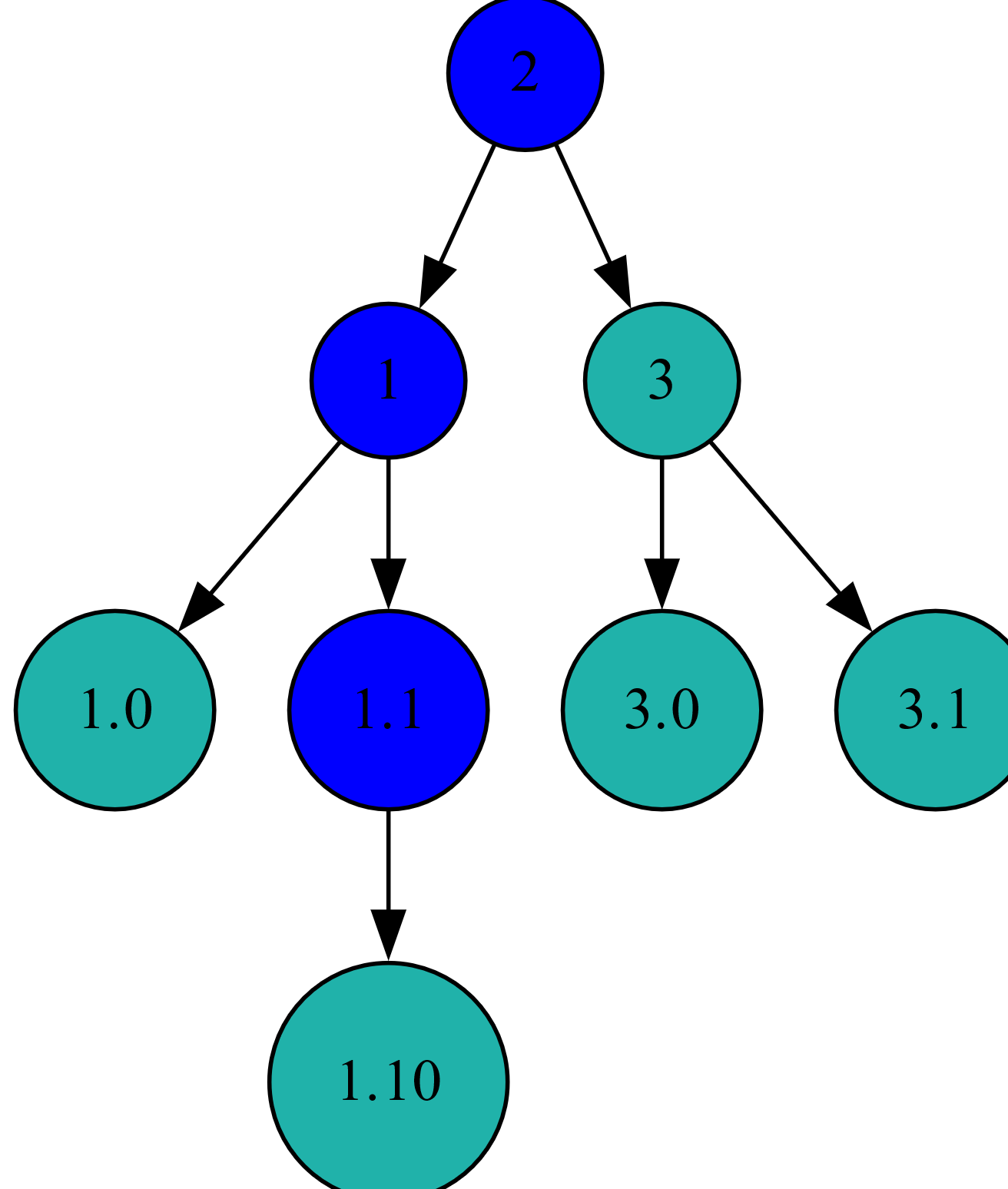
The blue nodes are currently running the isUnival function



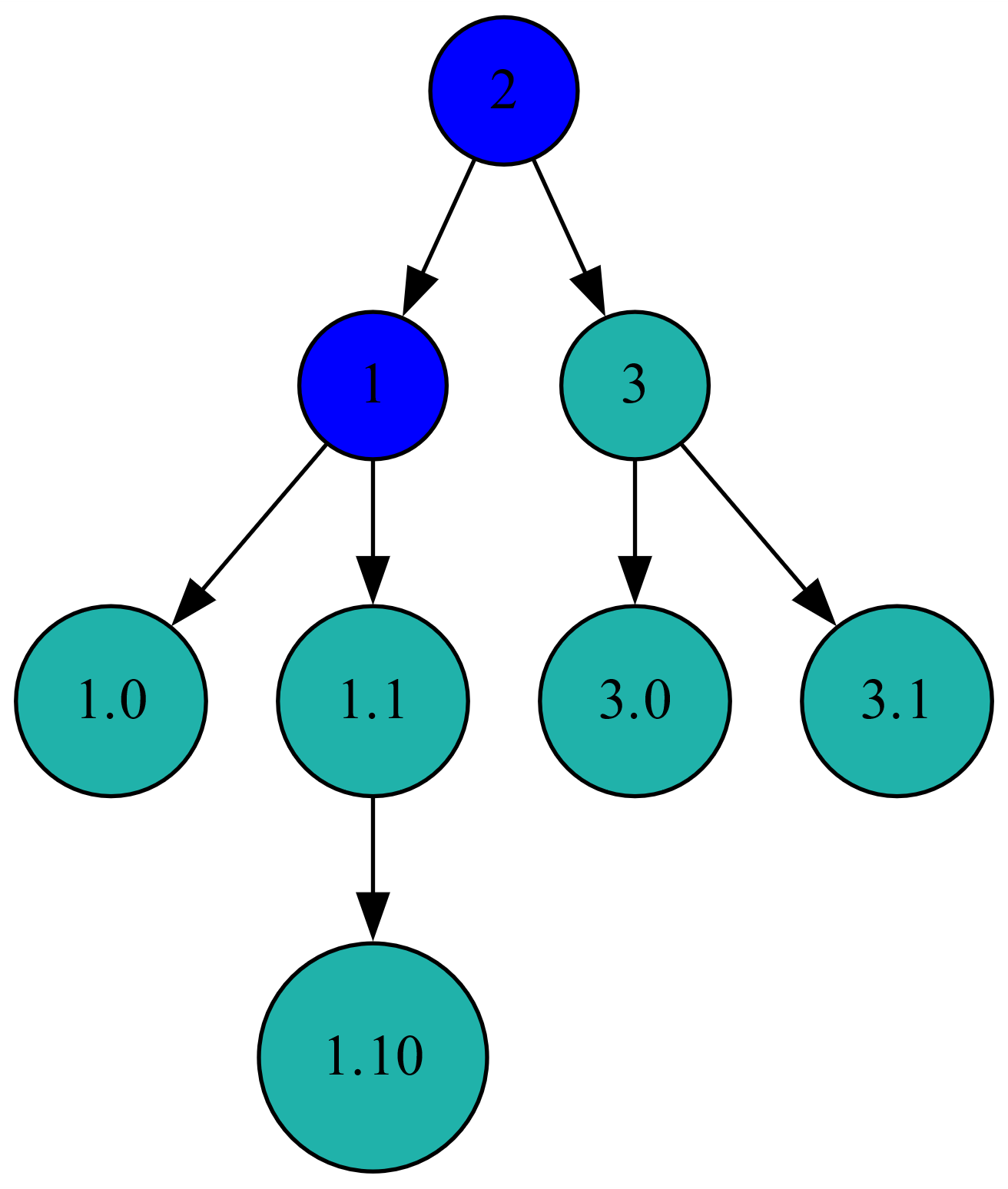
The isUnival function calls the function on every subtree



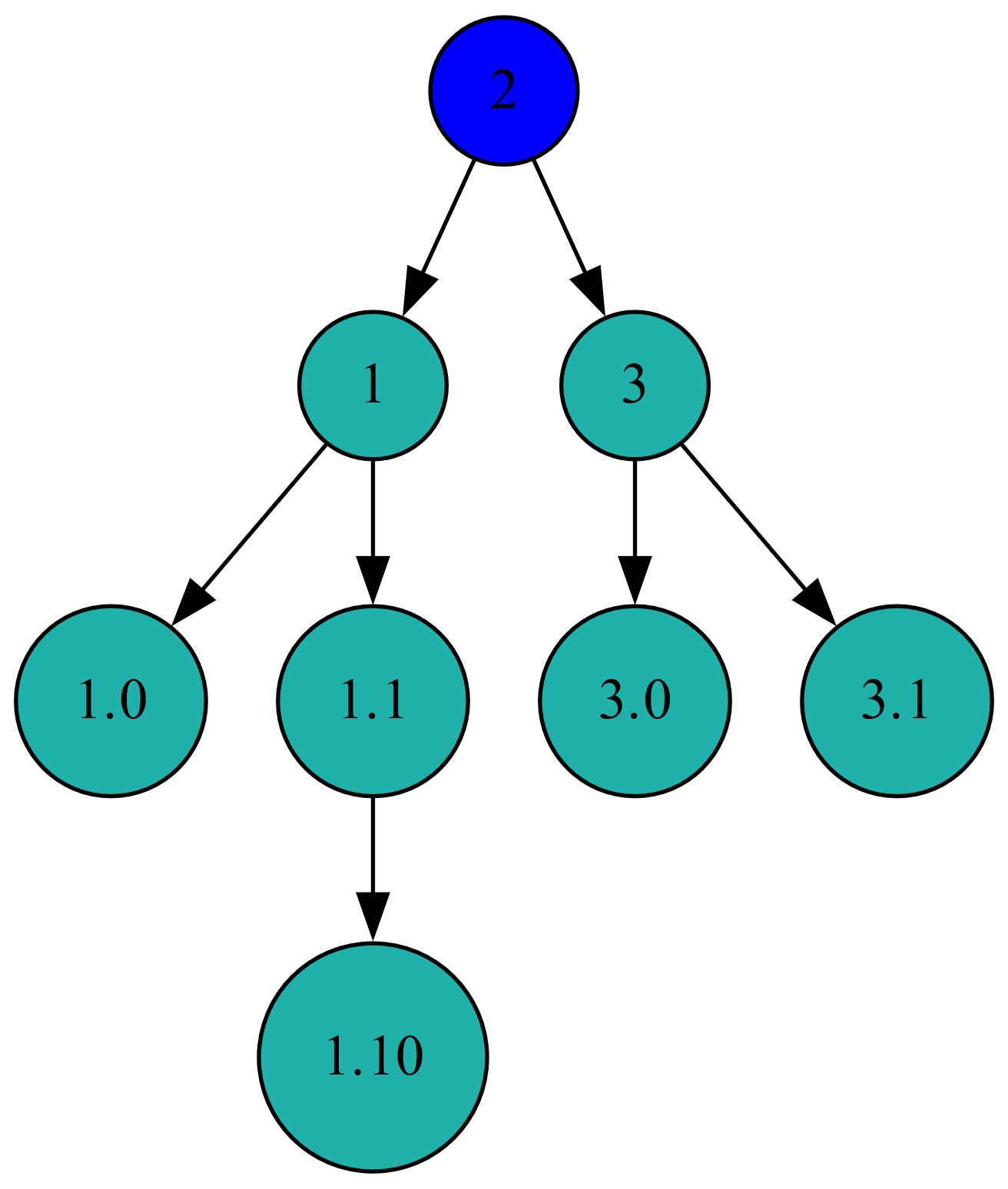
The leaf nodes return a true value



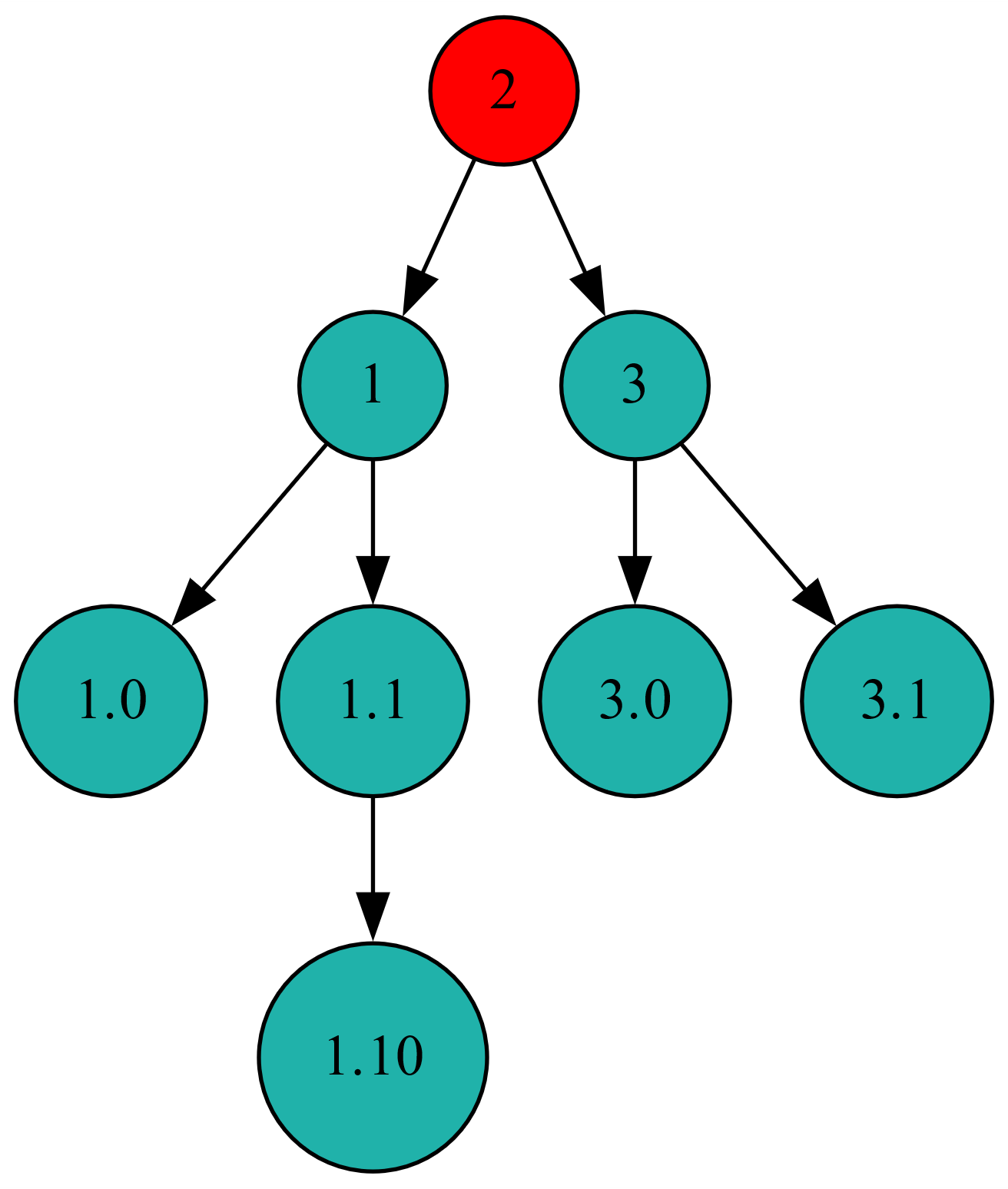
The 3 subtree returns true when its children both return true and are equal to the parent



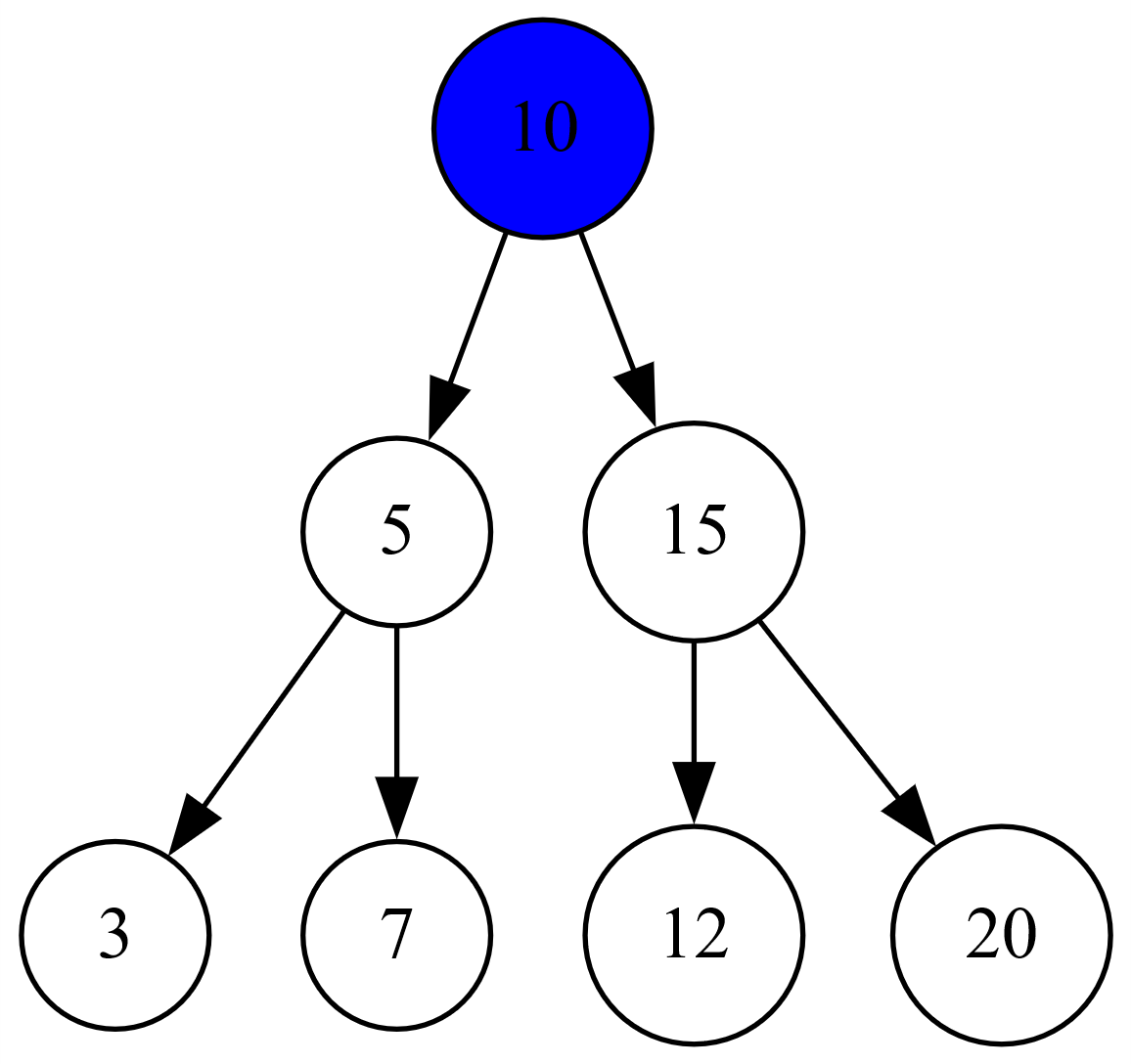
The 1.1 subtree returns true when its only child returns true



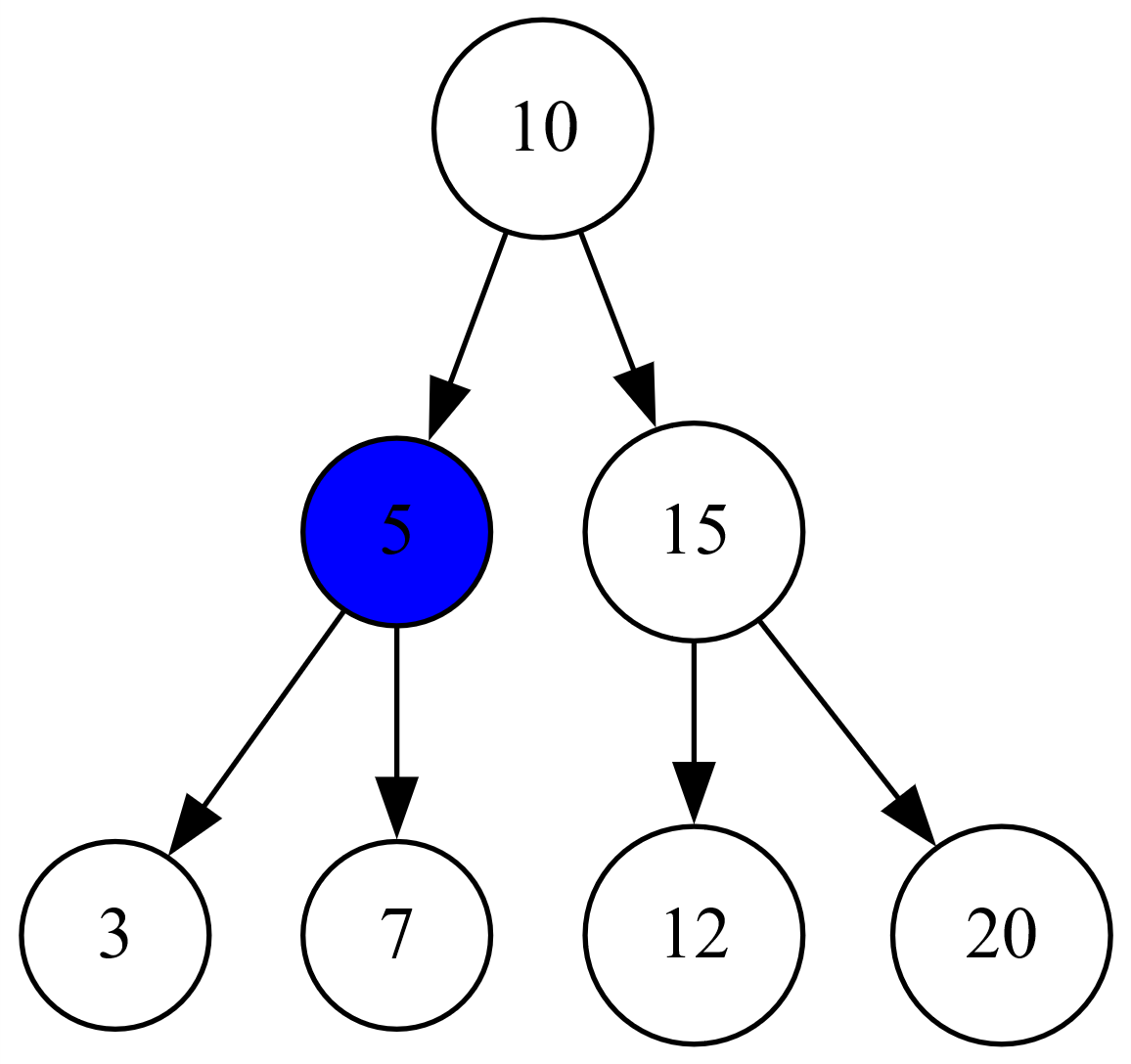
Each node that returns true increments the counter variable



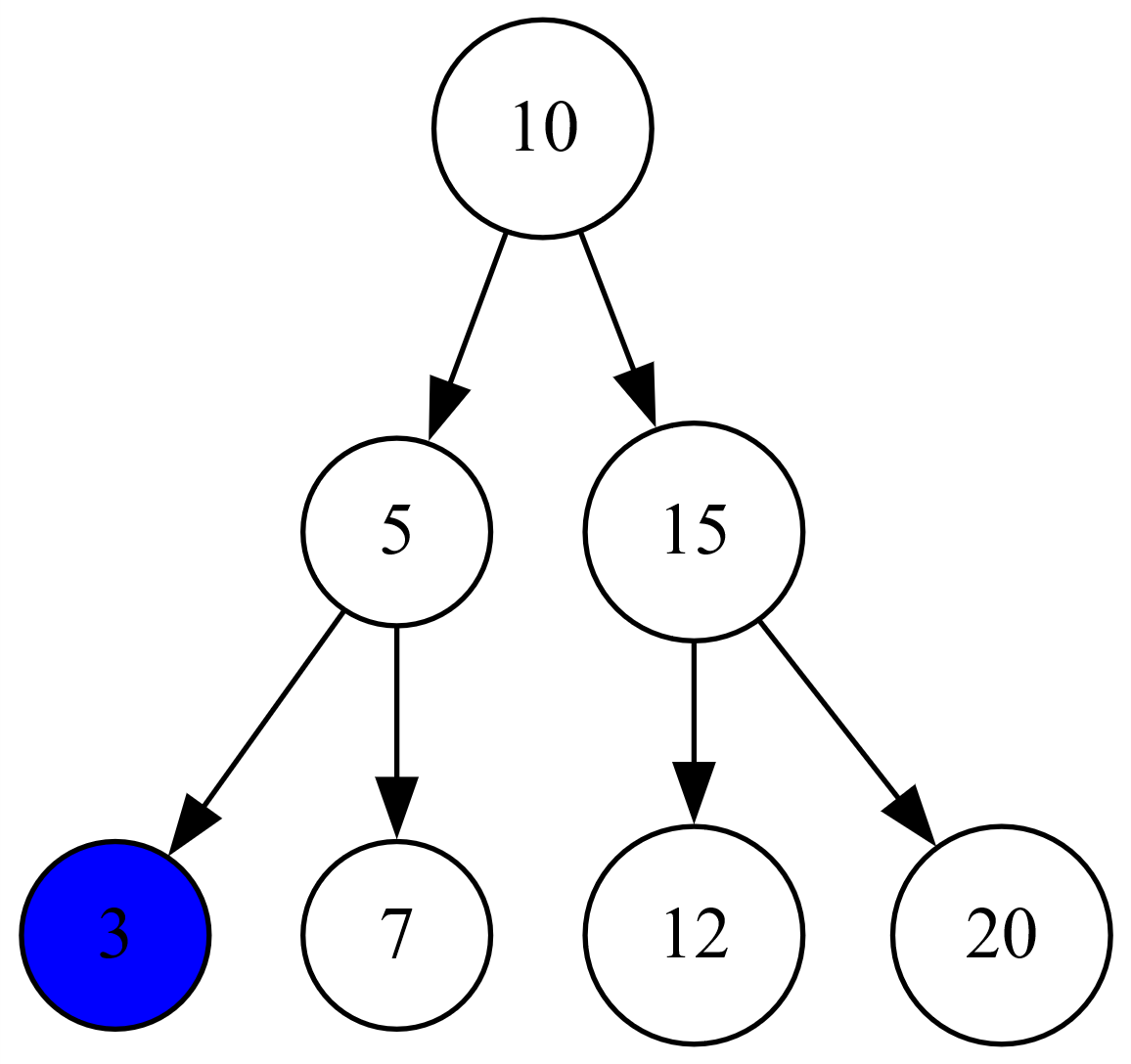
The root node returns false as its not equal to its children, though they returned true



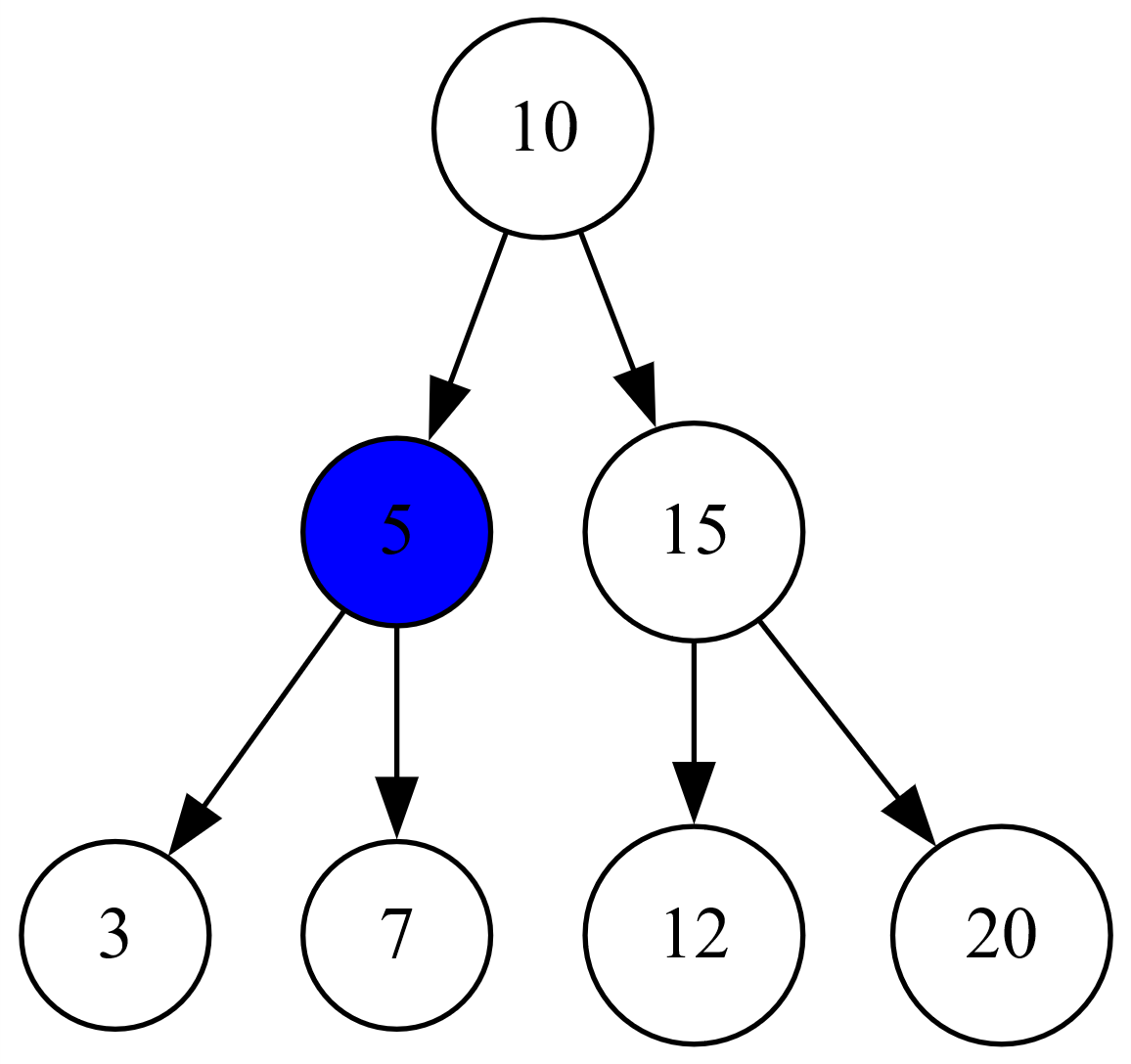
The coloured nodes are running the findMin function



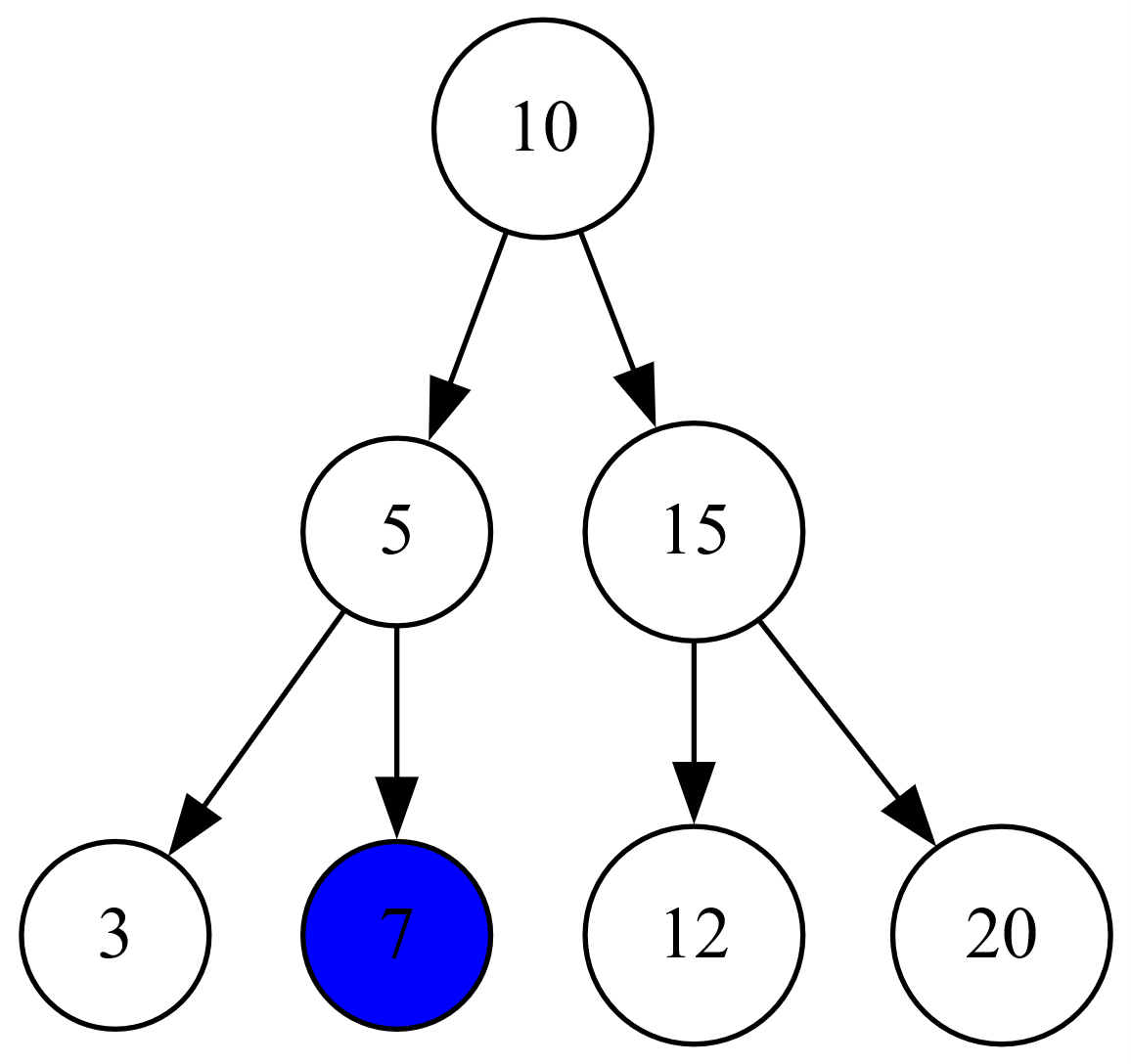
This function simply moves to the left



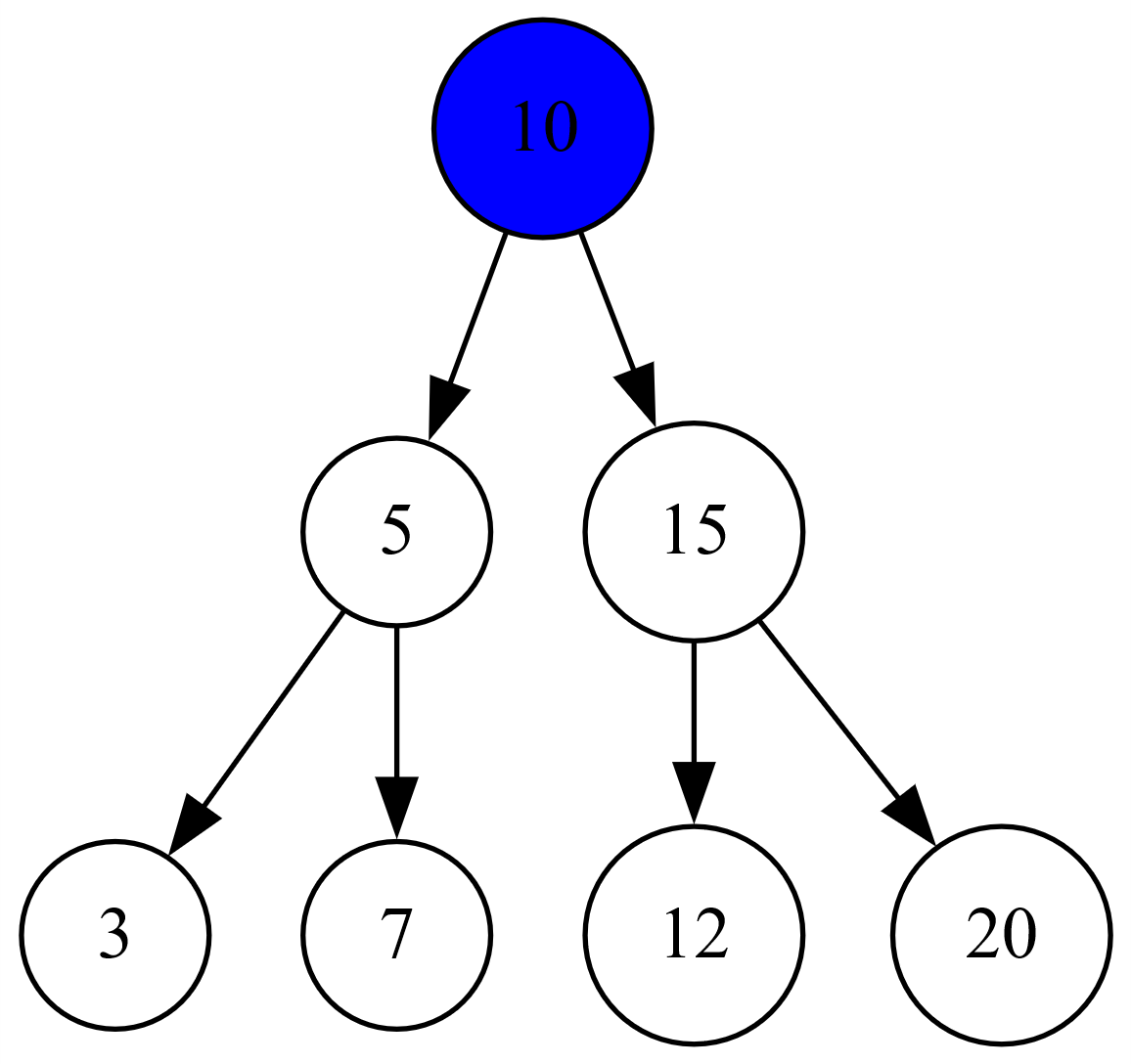
When it reaches a leaf node, it returns it as the smallest node

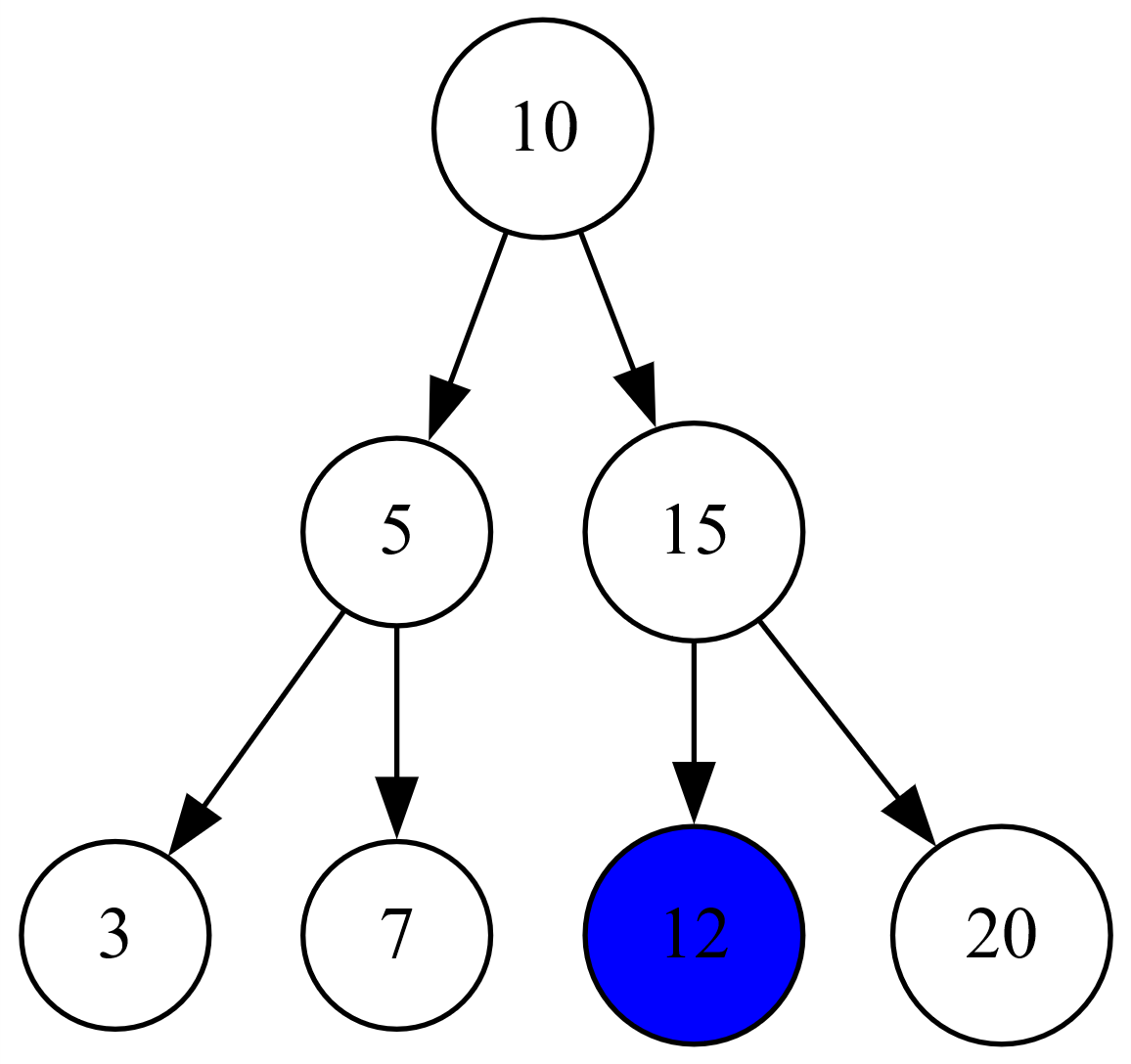


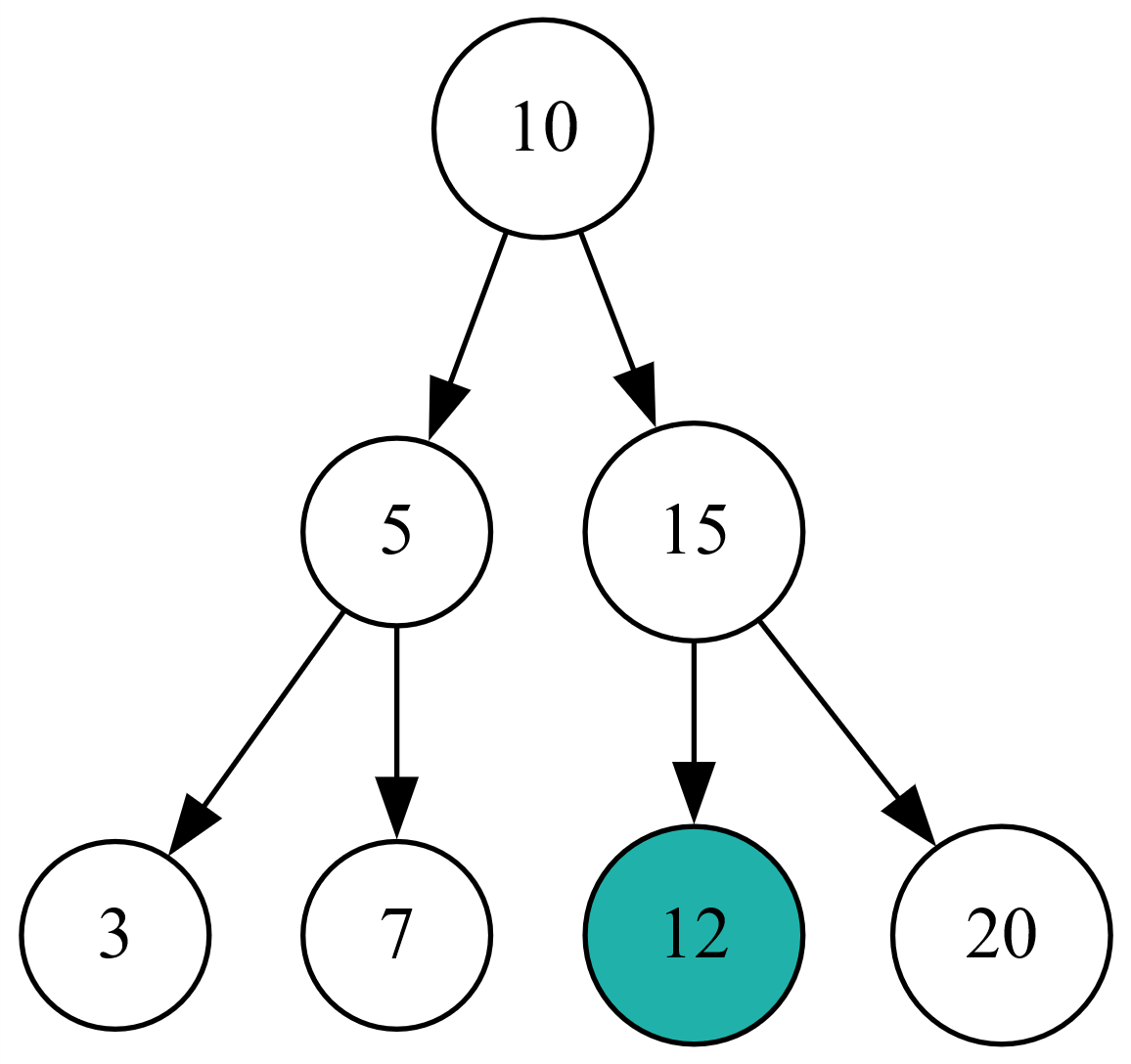
Here the coloured nodes are the inorder successors of the node in the previous image



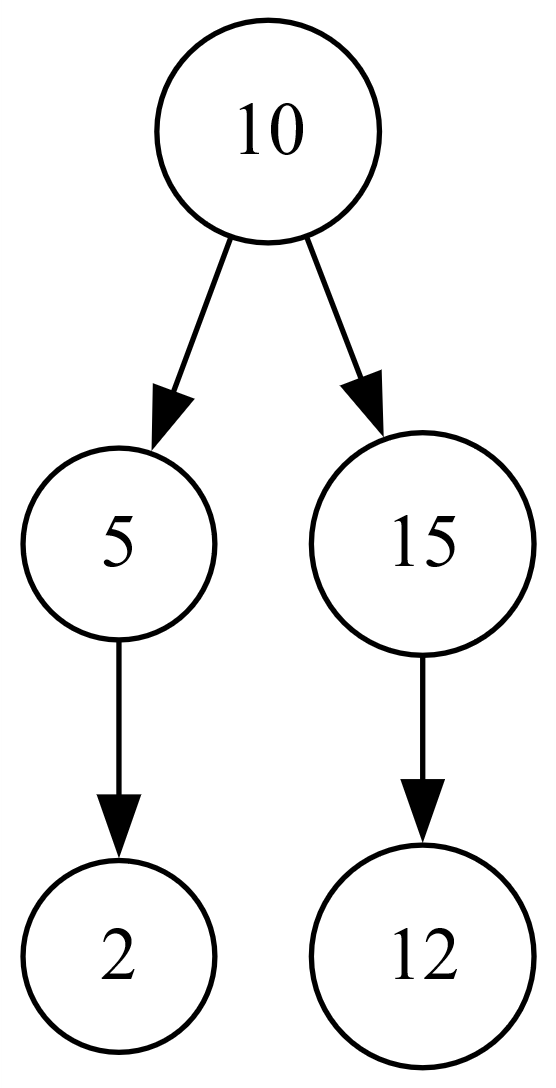
The inorderSuccessor function is called k-1 times(4) and returns that node







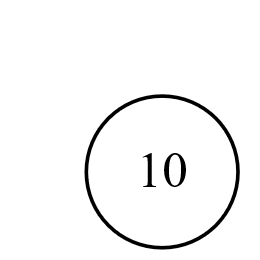
This node is returned



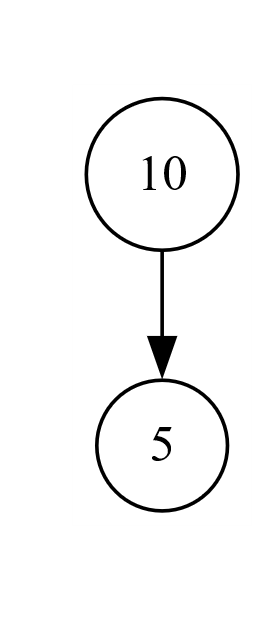
This is the original tree



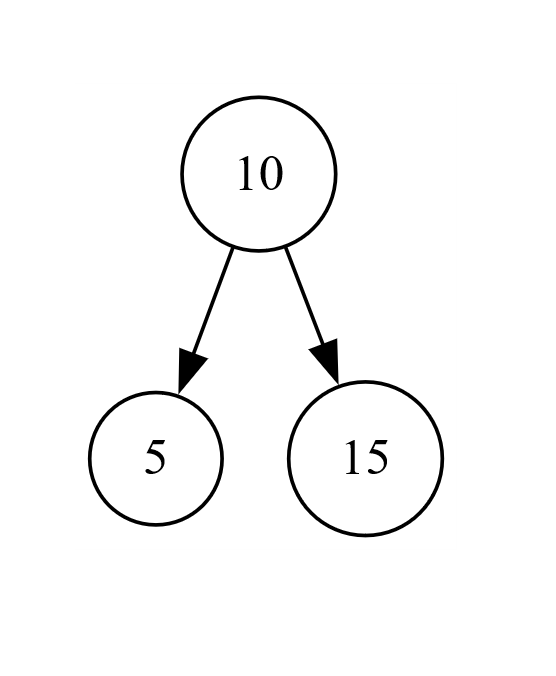
This is the serialized version with null elements

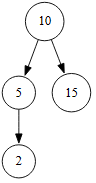


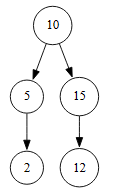
The deserializeBST function first creates a node with the first element



It then iterates through the elements and inserts them into the tree







This is the tree returned by the deserializeBST function