```
# I will assume that you already know how to create a node for a tree, so I won't cover that
# I also won't cover inorder traversal

def find_end(arr):
        open = close = end = 0 # Set all to zero
        finish = false # Could also use a do-while loop but that doesn't exist in Python
        # The reason you need to do this is to prevent the loop from not happening when
        # open and close are both zero.
        while not finish:
                if arr[ind] == "{", open += 1
                elif arr[ind] == "}", close += 1
                if open == close, finish = true
                ind += 1
        return ind

def parse(arr):
        arr.pop(0) # Should be {
        if arr[0] == "}": # Implies empty node
                arr.pop(0)
                return Node(None)
        else:
                arr.pop(len(arr)-1) # Get rid of the } at the end
                placeholder = 0 # Some numbers have multiple digits
                while arr[0] is a digit, placeholder = 10 * placeholder + int(arr.pop(0))
                root = Node(placeholder)
                left_end = find_end(arr)
                left = subarray of arr from 0 to left_end-1
                right = subarray of arr from left_end to len(arr)-1
                root.left = parse(left)
                root.right = parse(right)
                return root

# In the main function, you should take the input string, purge it of all spaces, and store
# each remaining character in an array.

root = parse(arr)
inorder(root)
```