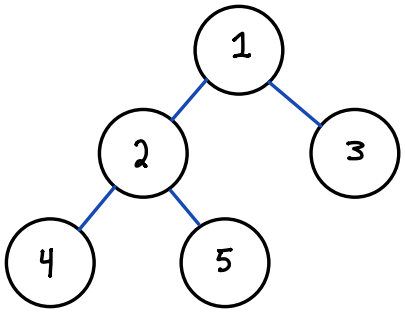
Size of a Binary Tree

Size of Tree:

The total number of nodes in the tree.

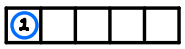


Stack data structure

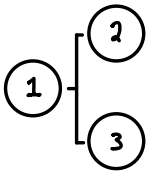
Step-by-Step:

Check if root node is not NULL

Start with tree node in the stack

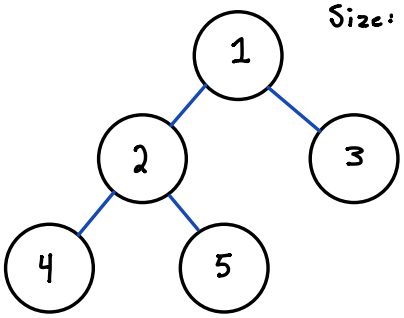


Then check if they have any children and push them on the stack as well



~ has two children

Size: 1

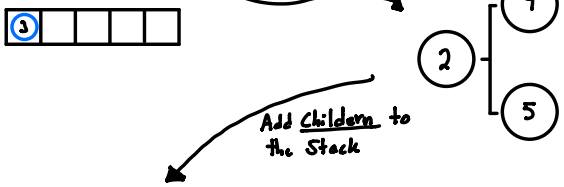


- Add Children to the stack

- increase counter by 2

Size = 1 + 2
Size = 3

Take top of the stack off: Check for Children



Add Children to the Stack

- increase counter by 2

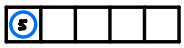
Size = 3 + 2
Size = 5

- Pop 3 off the stack

3

- Check for children
  - none

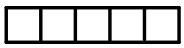
- Pop 4 off the stack



4

- Check for children
  - none

- Pop 5 off the stack



5

- Check for children
  - none

Size Function Code:

```
def size_(self, node):
    if node is None:
        return 0
    return 1 + self.size_(node.left) + self.size_(node.right)

def size(self):
    if self.root is None:
        return 0

    stack = Stack()
    stack.push(self.root)
    size = 1
    while stack:
        node = stack.pop()
        if node.left:
            size += 1
            stack.push(node.left)
        if node.right:
            size += 1
            stack.push(node.right)
    return size
```