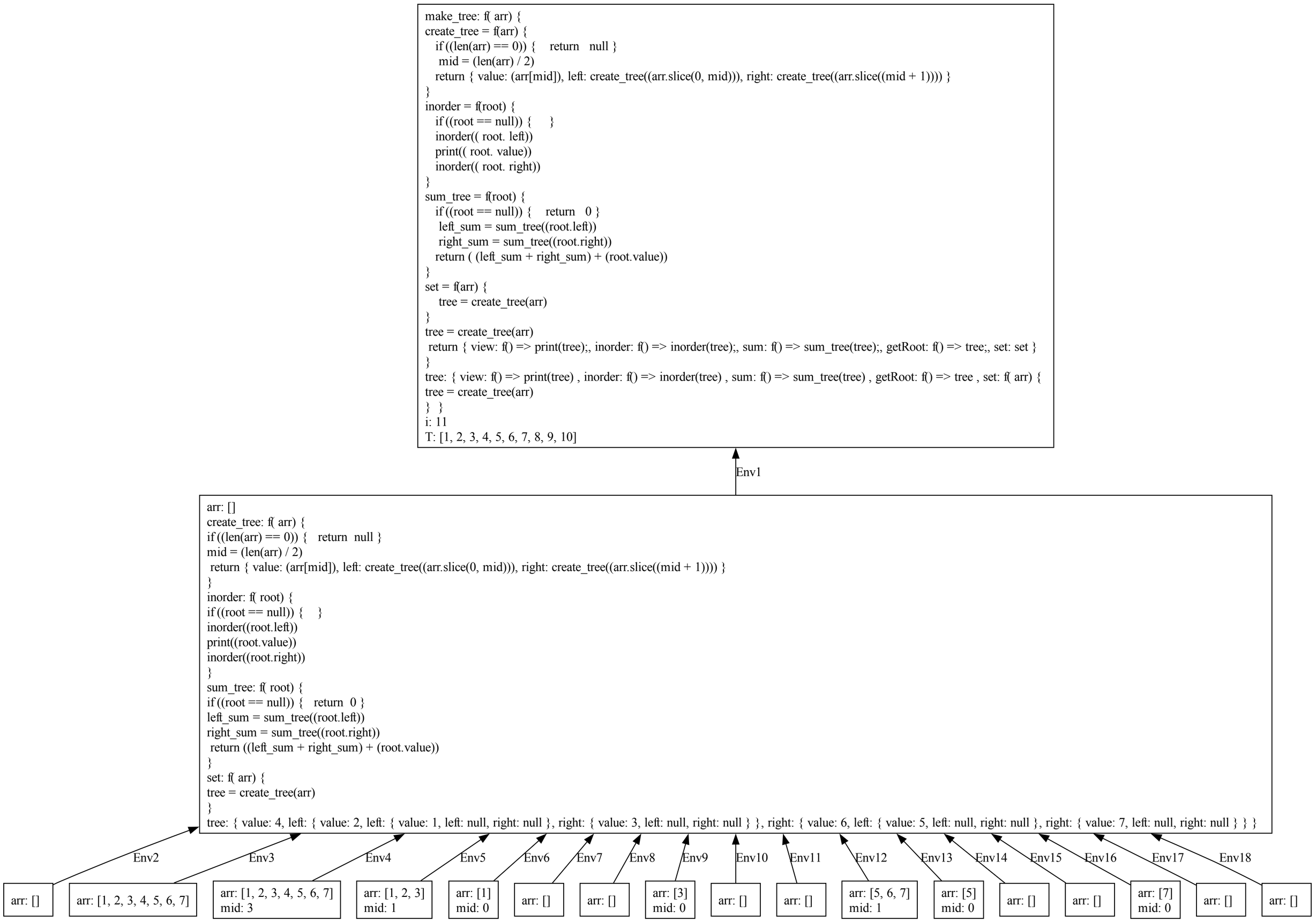
make_tree: f(arr) {
create_tree = f(arr) {
if ((len(arr) == 0)) { return null }
mid = (len(arr) / 2)
return { value: (arr[mid]), left: create_tree((arr.slice(0, mid))), right: create_tree((arr.slice((mid + 1)))) }
}
inorder = f(root) {
if ((root == null)) { }
inorder((root. left))
print((root. value))
inorder((root. right))
}
sum_tree = f(root) {
if ((root == null)) { return 0 }
left_sum = sum_tree((root.left))
right_sum = sum_tree((root.right))
return ((left_sum + right_sum) + (root.value))
}
set = f(arr) {
tree = create_tree(arr)
}
tree = create_tree(arr)
return { view: f() => print(tree);, inorder: f() => inorder(tree);, sum: f() => sum_tree(tree);, getRoot: f() => tree;, set: set }
}
tree: { view: f() => print(tree) , inorder: f() => inorder(tree) , sum: f() => sum_tree(tree) , getRoot: f() => tree , set: f(arr) {
tree = create_tree(arr)
} }
i: 11
T: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Env1
arr: []
create_tree: f(arr) {
if ((len(arr) == 0)) { return null }
mid = (len(arr) / 2)
return { value: (arr[mid]), left: create_tree((arr.slice(0, mid))), right: create_tree((arr.slice((mid + 1)))) }
}
inorder: f(root) {
if ((root == null)) { }
inorder((root.left))
print((root.value))
inorder((root.right))
}
sum_tree: f(root) {
if ((root == null)) { return 0 }
left_sum = sum_tree((root.left))
right_sum = sum_tree((root.right))
return ((left_sum + right_sum) + (root.value))
}
set: f(arr) {
tree = create_tree(arr)
}
tree: { value: 4, left: { value: 2, left: { value: 1, left: null, right: null }, right: { value: 3, left: null, right: null } }, right: { value: 6, left: { value: 5, left: null, right: null }, right: { value: 7, left: null, right: null } } }
Env2
Env3
Env4
Env5
Env6
Env7
Env8
Env9
Env10
Env11
Env12
Env13
Env14
Env15
Env16
Env17
Env18
arr: []
arr: [1, 2, 3, 4, 5, 6, 7]
arr: [1, 2, 3, 4, 5, 6, 7]
mid: 3
arr: [1, 2, 3]
mid: 1
arr: [1]
mid: 0
arr: []
arr: []
arr: [3]
mid: 0
arr: []
arr: []
arr: [5, 6, 7]
mid: 1
arr: [5]
mid: 0
arr: []
arr: []
arr: [7]
mid: 0
arr: []
arr: []