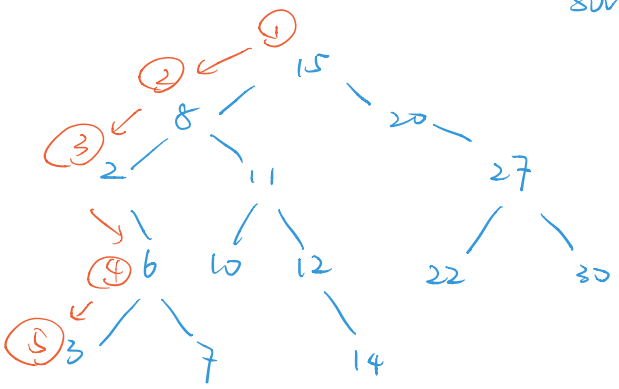
In order traversal:

if (localRoot != null)

inorder(localRoot.leftChild)

print

inorder(localRoot.rightChild)



sorted != BST

from root: 15

inorder(15)

↓ left

inorder(8)

↓ left

inorder(2)

↓ left end

~~inorder(null)~~ base

↳ Done print 2 2

inorder(6)

↓ left

inorder(3)

↓ left end

~~inorder(null)~~ base

↳ Done print(3) print(6) 2, 3, 6

inorder(7)

↓

~~inorder(null)~~ end

↳ Done print(7) print(8) 2,3,6,7,8

inorder(11)
↓ left
inorder(10) end
↳ print(10) print(11) 2,3,6,7,8,10,11

inorder(12) end
↳ print(12) 2,3,6,7,8,10,11,12

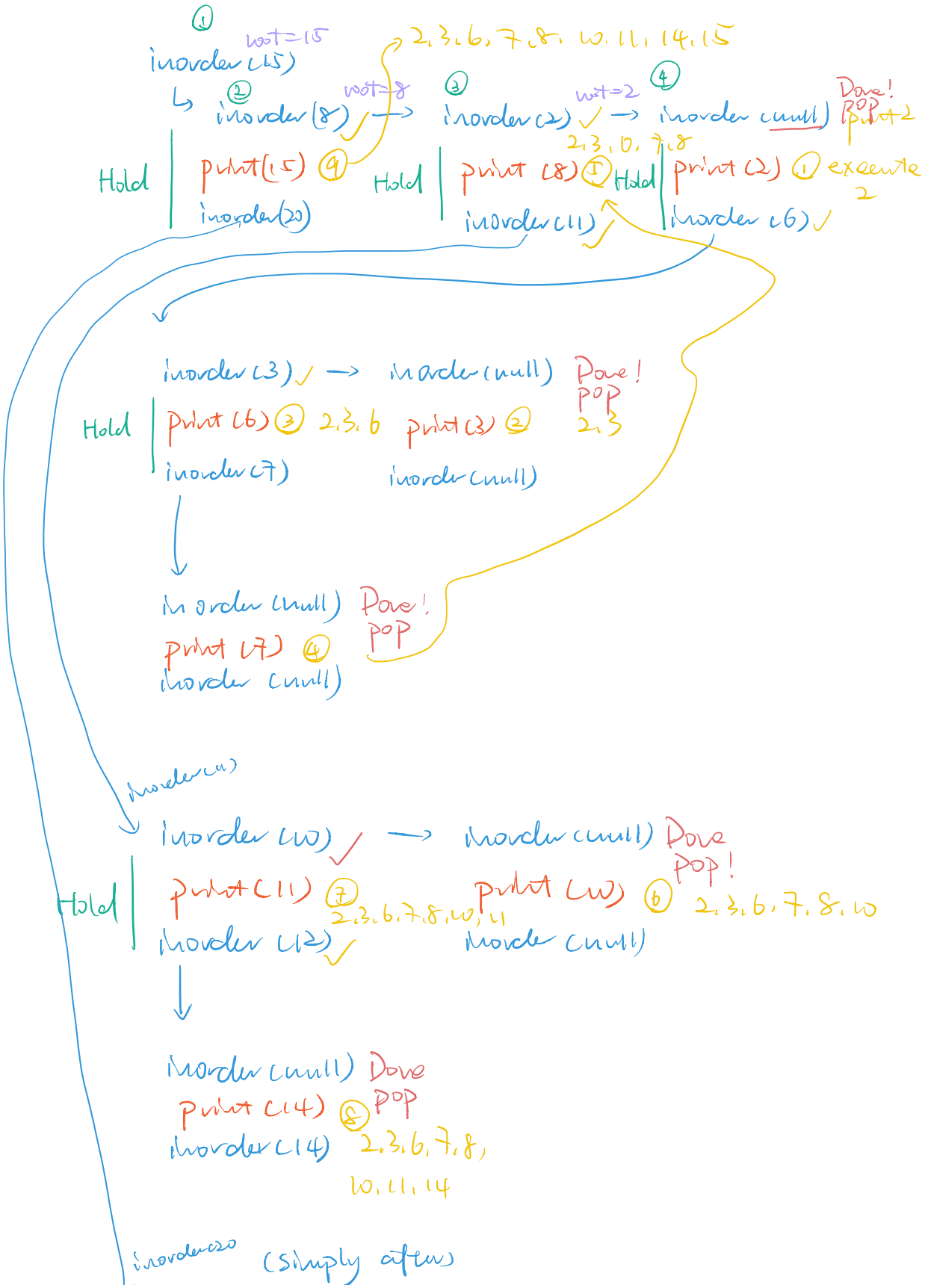
inorder(14) end 2,3,6,7,8,10,12,14,15
↳ print(14) . print(15)
inorder(20) end
↳ print(20) 2,3,6,7,8,10,12,14,15,20

inorder(27)
↓ left
2,3,6,7,8,10,12,14,15,20
22,27
inorder(22) end
↳ print(22) print(27)
inorder(30) end
↳ print(30)

Result:

2,3,6,7,8,10,11,12,14,15,20,22,27,30

More detailed here ↓


inorder(15)
root=15
inorder(8)
root=8
inorder(2)
root=2
inorder(null)
Done! POP
print 2
2,3,6,7,8,10,11,14,15
Hold
print(15)
inorder(20)
print(8)
inorder(11)
print(2)
execute 2
inorder(6)
inorder(3)
inorder(null)
print(6)
2,3,6
print(3)
2,3
inorder(7)
inorder(null)
print(7)
inorder(null)
inorder(10)
inorder(10)
inorder(null)
print(11)
print(10)
2,3,6,7,8,10,11
inorder(12)
inorder(null)
inorder(null)
print(14)
2,3,6,7,8,10,11,14
inorder(20)
(simply after)

inorder(20)

print(20) (10)

inorder(27)

inorder(22)

print(22) (11)

print(27) (12)

inorder(30)

print(30) (13)

Output: 2, 3, 6, 7, 8, 10, 11, 12, 14, 15, 20, 22, 27, 30