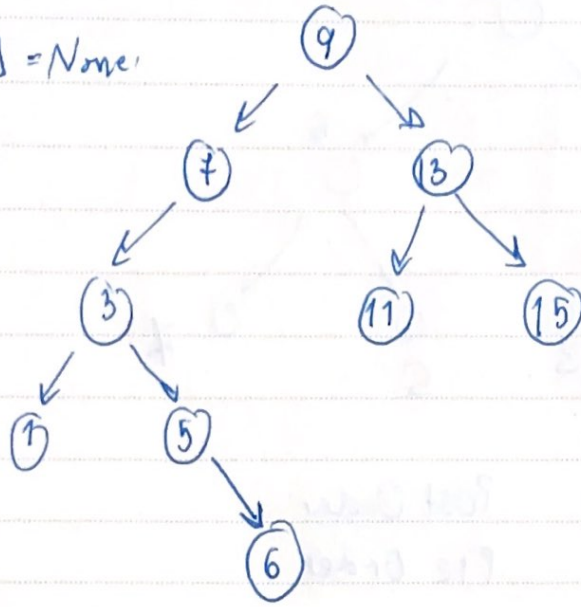
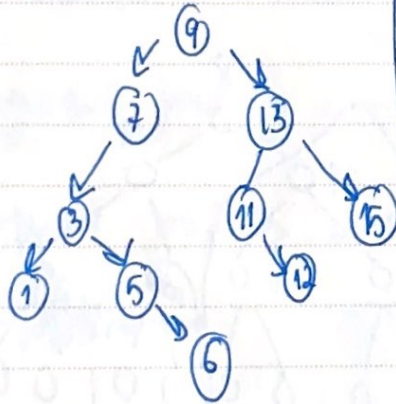


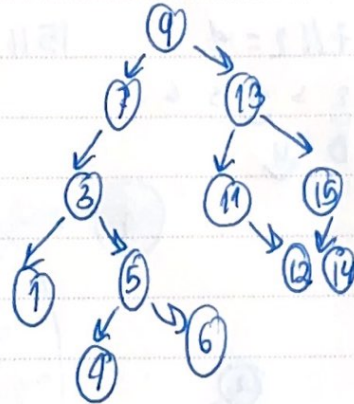
① bst [6] = None



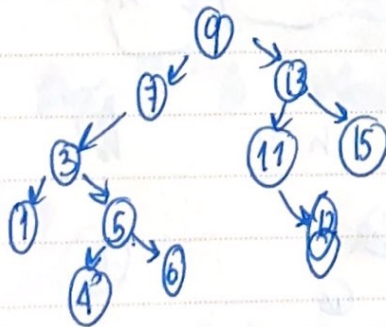
② bst [12] = None



④ bst [14]

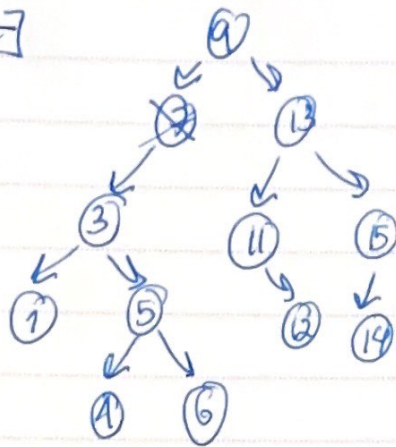


③ bst [4] = None

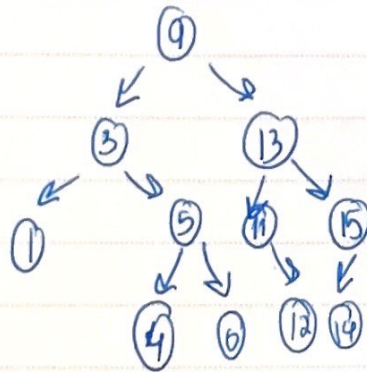


del bst [7]

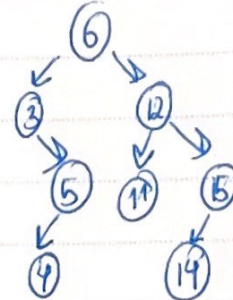
tree:



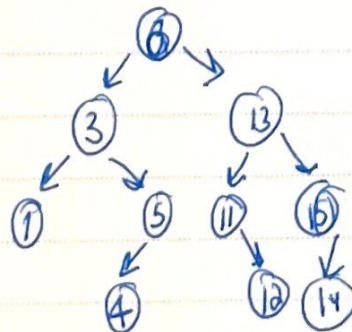
1) del bst [7]:



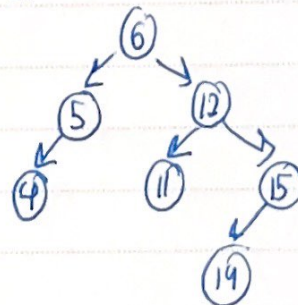
4) del bst [1]



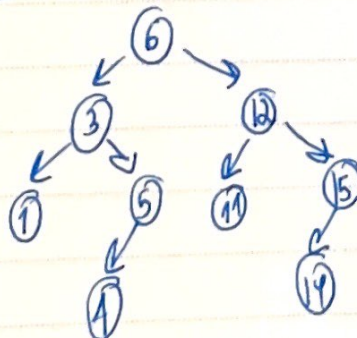
2) del bst [0]



5) del bst [3]



3) del bst [13]



Q2 c:

Part a

1st iteration: 1 operation

2nd iteration: 2 operation

⋮

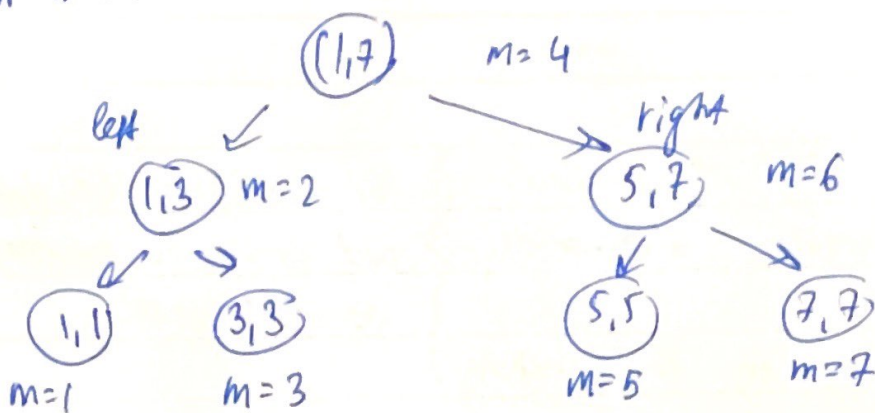
nth iteration: n operation

} since setitem uses find-node
and find-node traverses
the whole tree

$$1 + 2 + \dots + n = \Theta(n^2) + c = \Theta(n^2)$$

for $n=7$:

add-items:

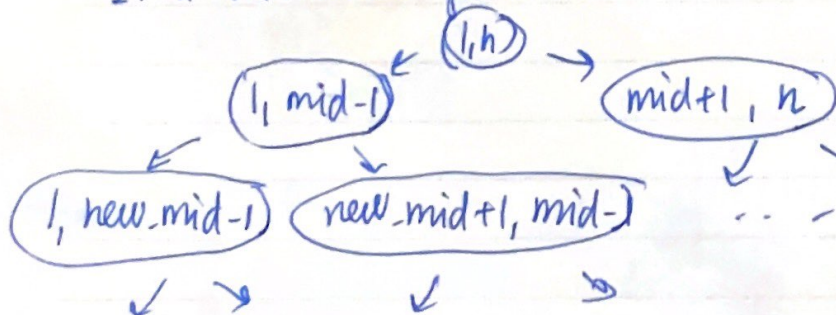


each recursive call creates a node with the middle element and makes ~~more~~ 2 recursive calls excluding the mid.

also each call makes constant work:

- by
- creating a new item, new node
 - setting node's left, right
 - left and right's parent.

In a generalized format



Level	# calls	total work
0	2	2
1	4	4
2	8	8
...
k	2^k	2^k
$\log_2 n$	n	n

(1,1)

$$1 + 2 + 4 + 8 + \dots + n = \Theta(n)$$

each row does

k^{th} row does 2^k work

there are $\log n$ rows (complete tree)

so $\Theta(n)$ work in total.