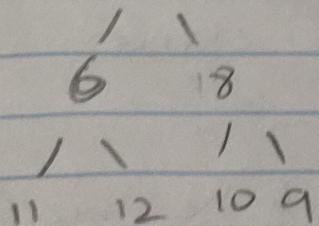
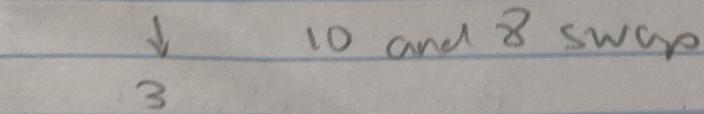
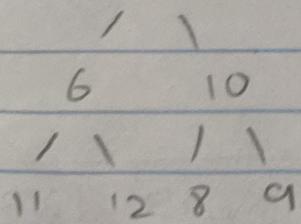
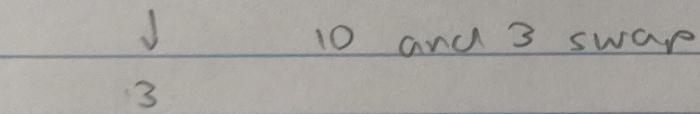
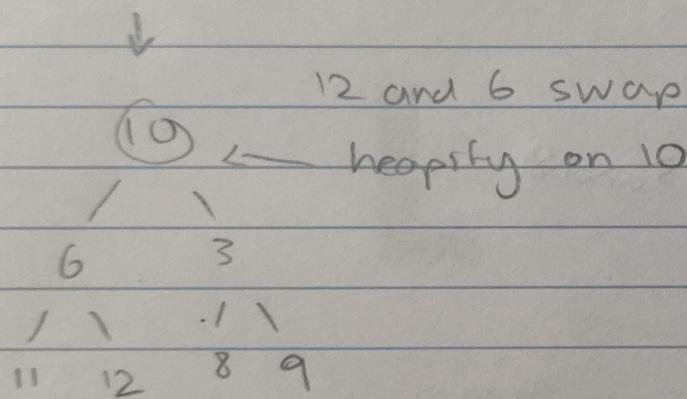
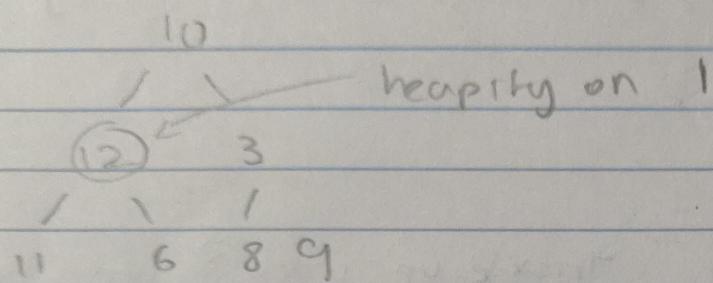
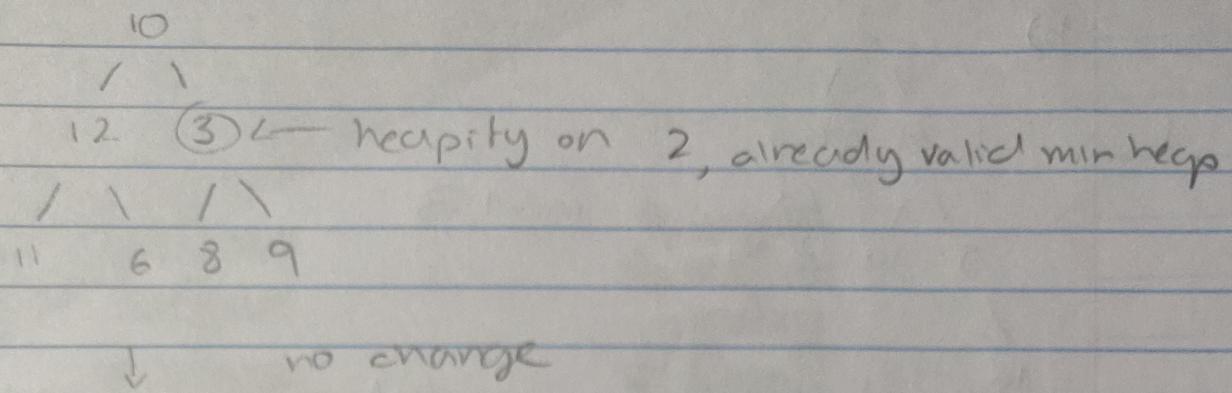
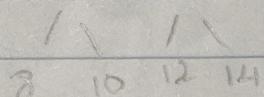
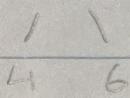


#4 a)



Final array: [3, 6, 8, 11, 12, 10, 9]

b) 2



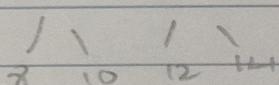
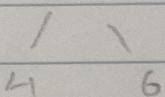
insert 3

16



3 is inserted at end of array

2

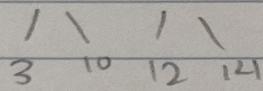
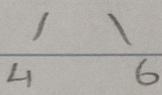


Trickle up index 8 until heap is fixed

16 3

↓ swap 3 and 8

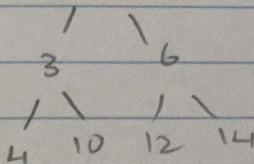
2



16 8

↓ swap 3 and 4

2



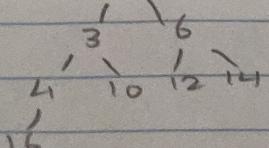
Tree after insert 3: [2, 3, 6, 4, 10, 12, 14, 16, 8]

Pop top element, replace 0th index with last element

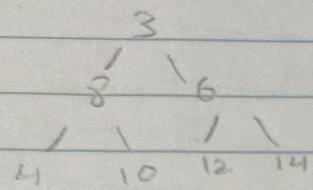
16 8



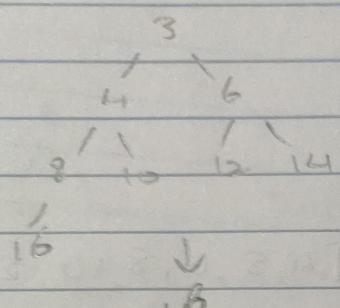
put 8 at the top of the tree, trickle down 0th index until heap is fixed



↓ swap 3 and 8

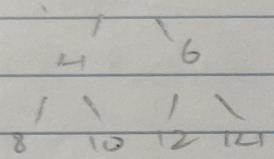
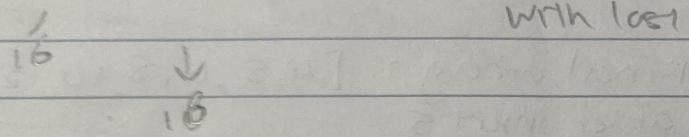


↓ swap 8 and 4



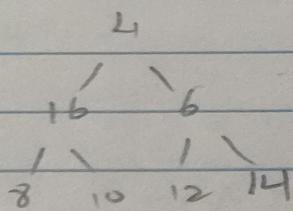
Tree after pop = [3, 4, 6, 8, 10, 12, 14, 16]

Pop top element, replace 0th index
with last element

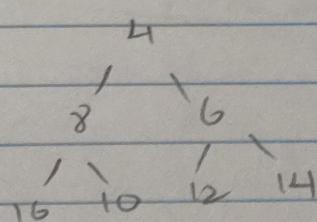


put 16 at the top of the tree,
trickle down 0th index until
heap is fixed

↓ swap 4 and 16

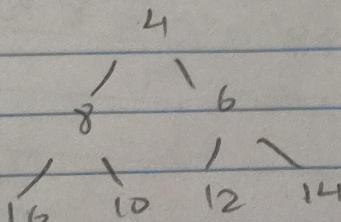


↓ swap 16 and 8



Tree after pop: 4, 8, 6, 16, 10, 12, 14
Insert 5

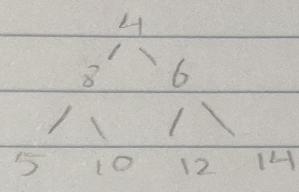
↓ 5 is inserted at when end of array



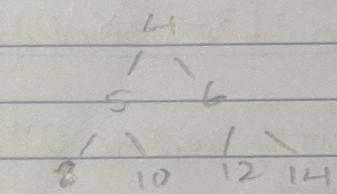
Trickle up index 7 until
heap is fixed

↓

↓ swap 16 and 5



↑ swap 8 and 5



Final array: [4, 5, 6, 8, 10, 12, 14, 16]
after insert 5