

Operations on AVL Tree - Part I

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Data Structure

Heap

By: Vishvadeep Gothi

Heap

A specialized tree-based data structure which is essentially ~~an almost~~ complete tree that satisfies the heap property.

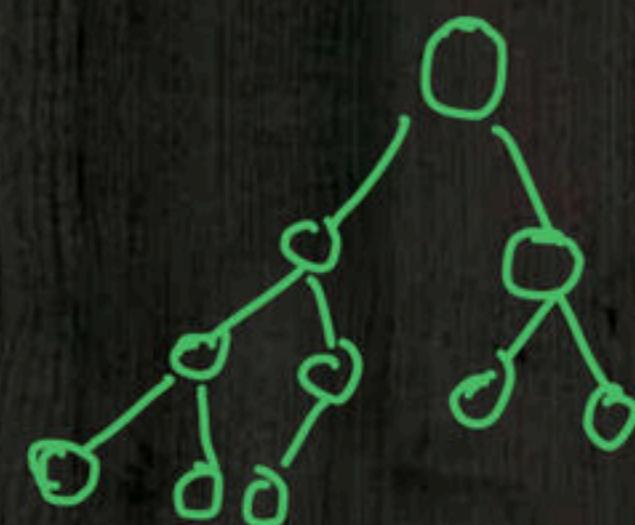


it's a priority base data structure.

Heap

2 Properties to be satisfied:

1. It should be a complete binary tree
2. Max heap or min heap property



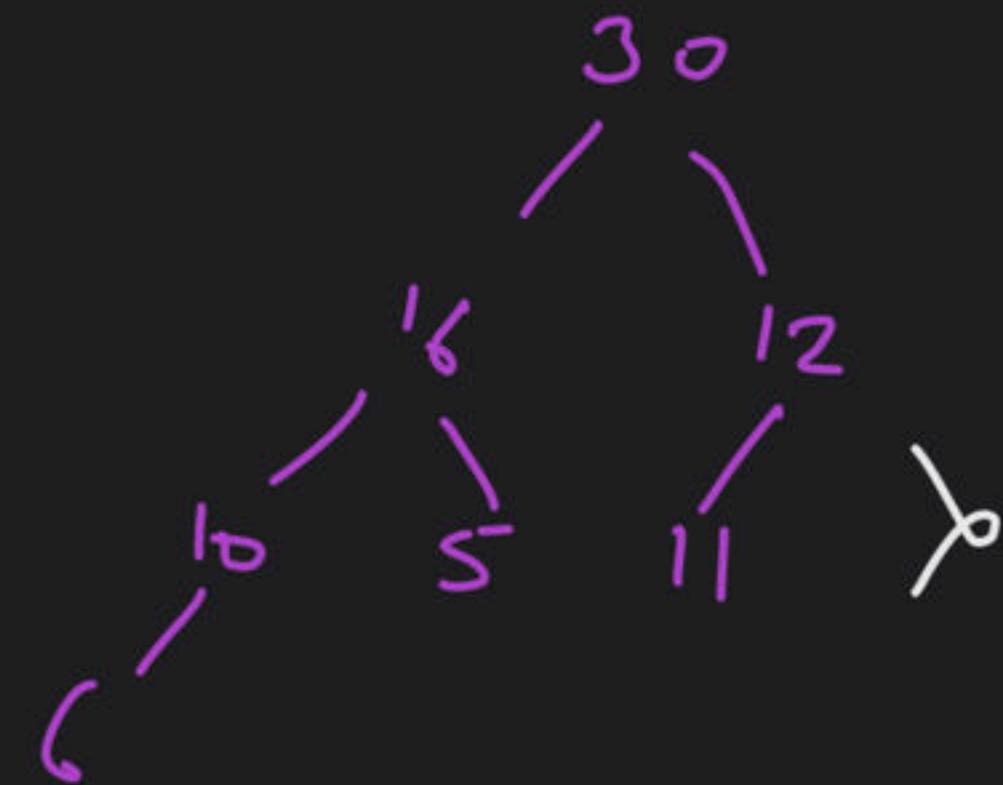
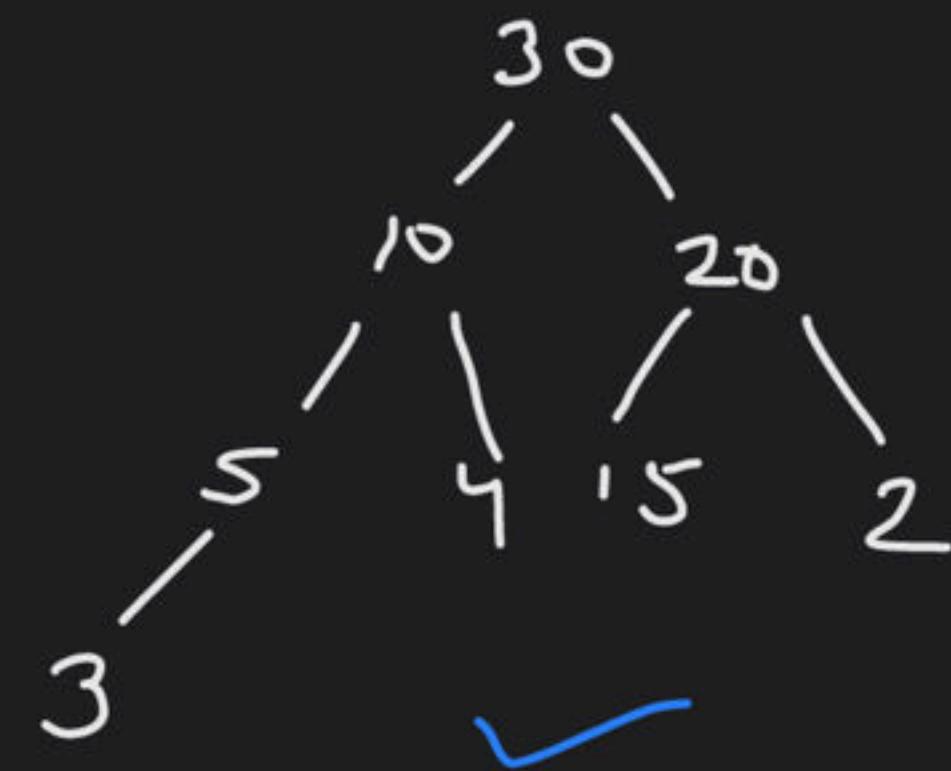
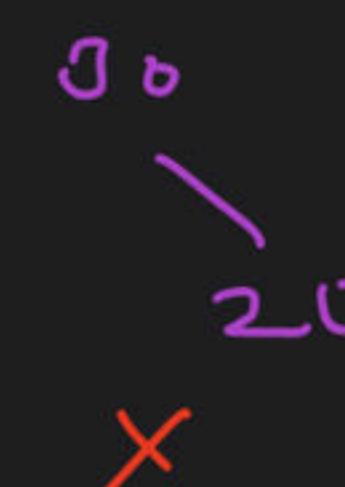
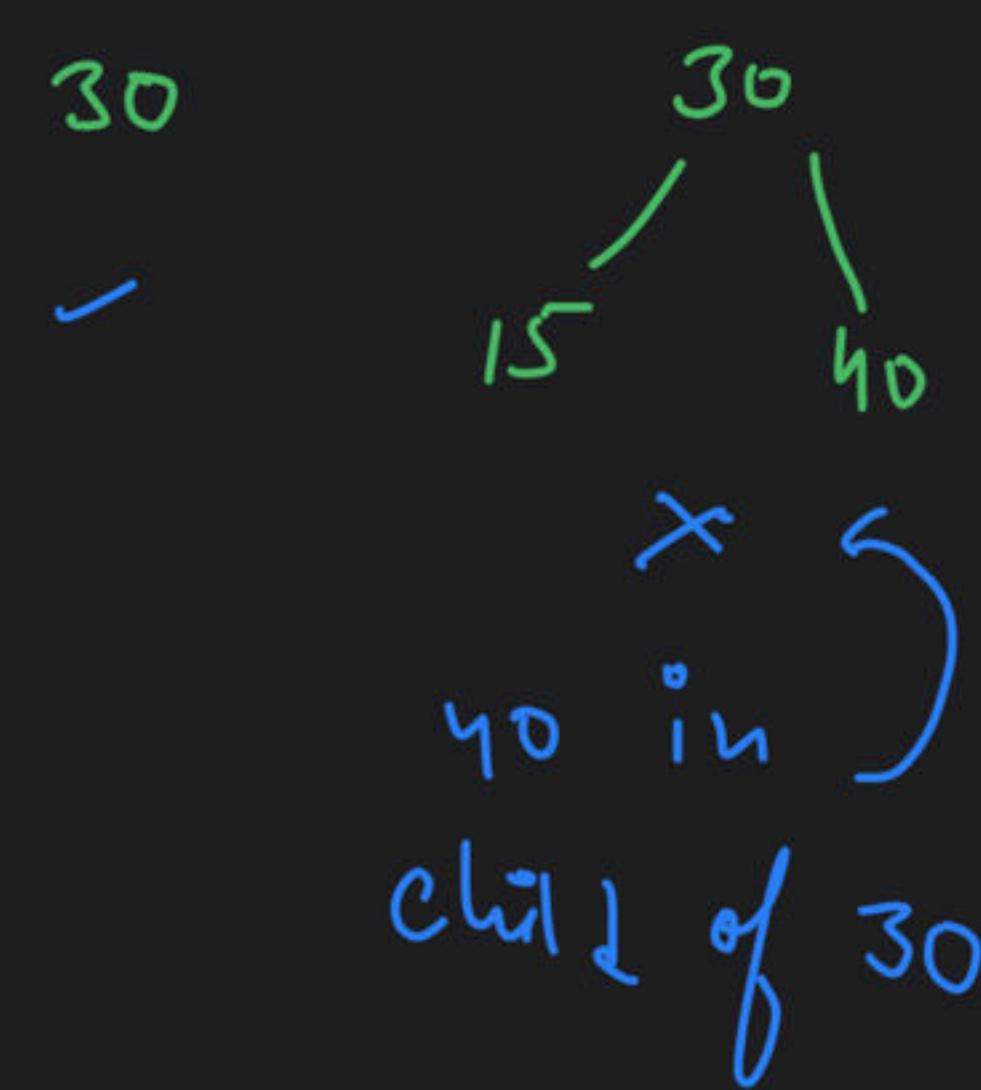
max heap :-

Every node should contain greater element than both of it's children.

min heap :-

Every node should contain lesser element than both of it's children.

Max heap:-



Array Representation of Heap

Similar to Array representation of CBT

Root stored at index 1

for any at index i ,

left child $\Rightarrow 2^{\circ} i$

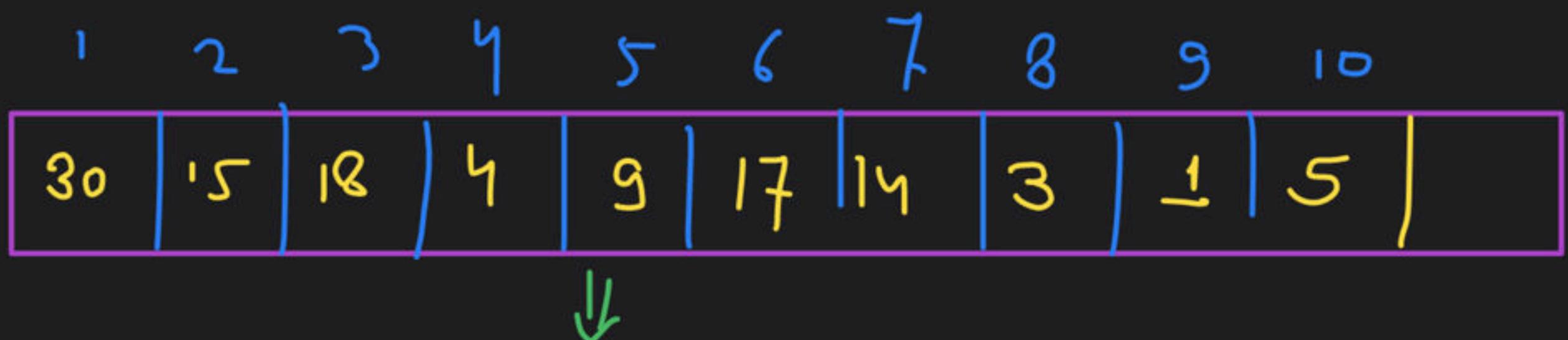
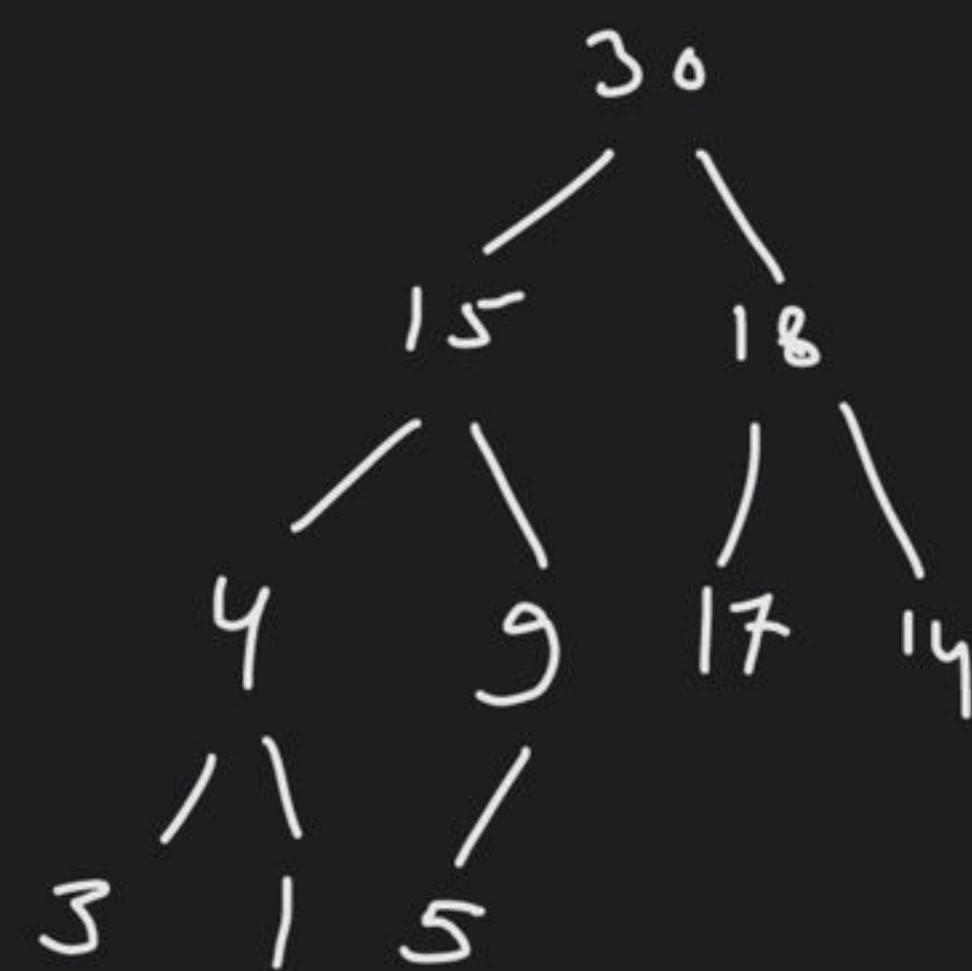
right child $\Rightarrow 2^{\circ} i + 1$

0

$\Rightarrow 2^{\circ} i + 1$

$\Rightarrow 2^{\circ} i + 2$

Ex:- max heap:-



30, 15, 18, 4, 9, 17, 14, 3, 1, 5

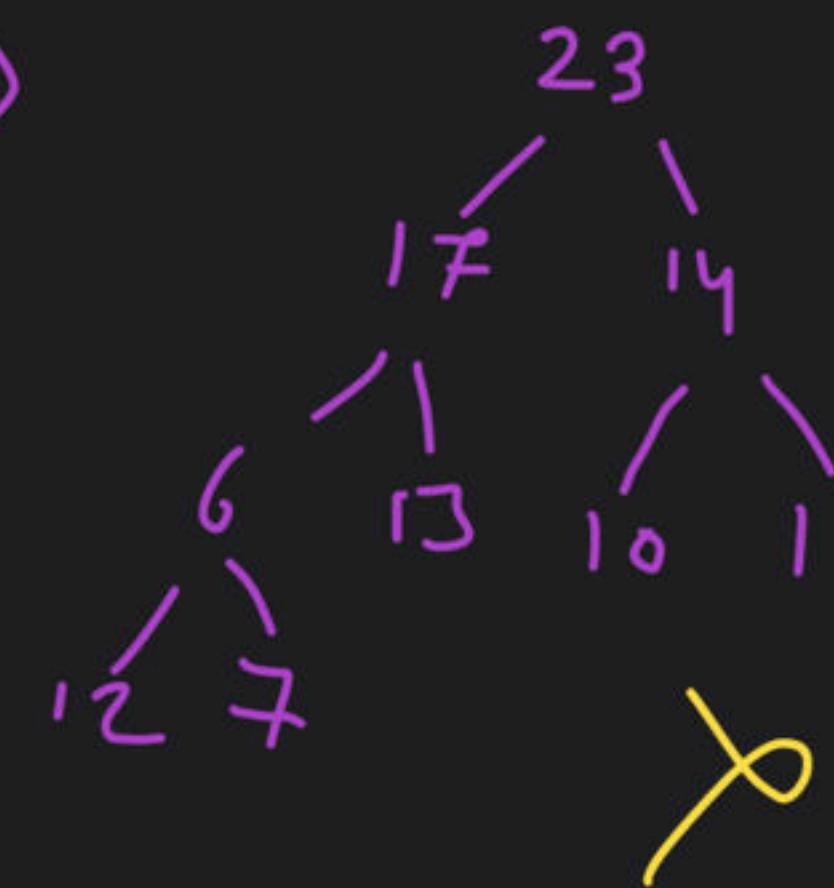
Ans] is the following array representation of heap
is representing a min heap or not

5, 10, 18, 20, 21, 19, 30, 31, 24, 23, 29

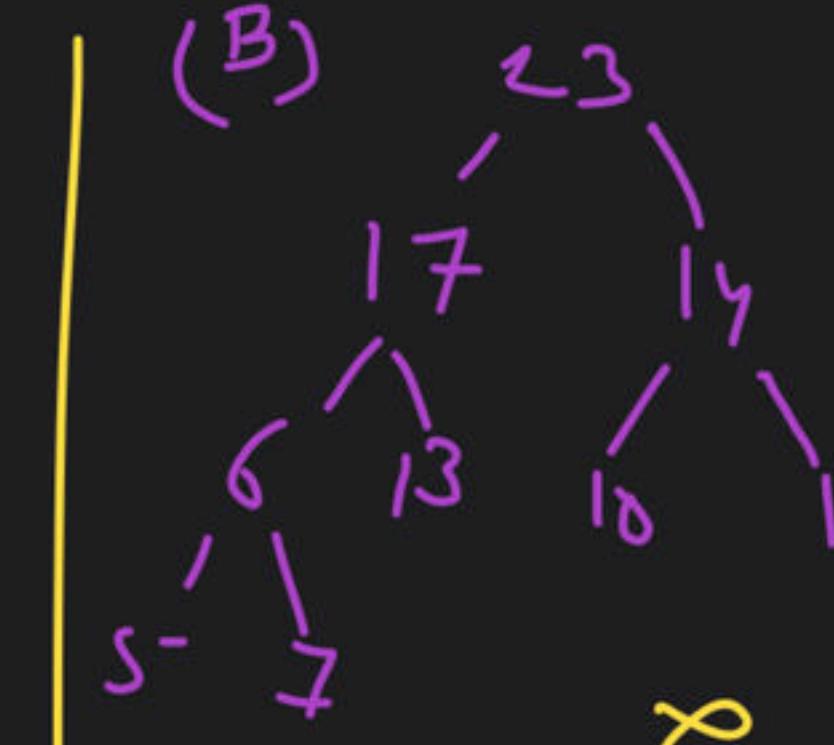


Yes

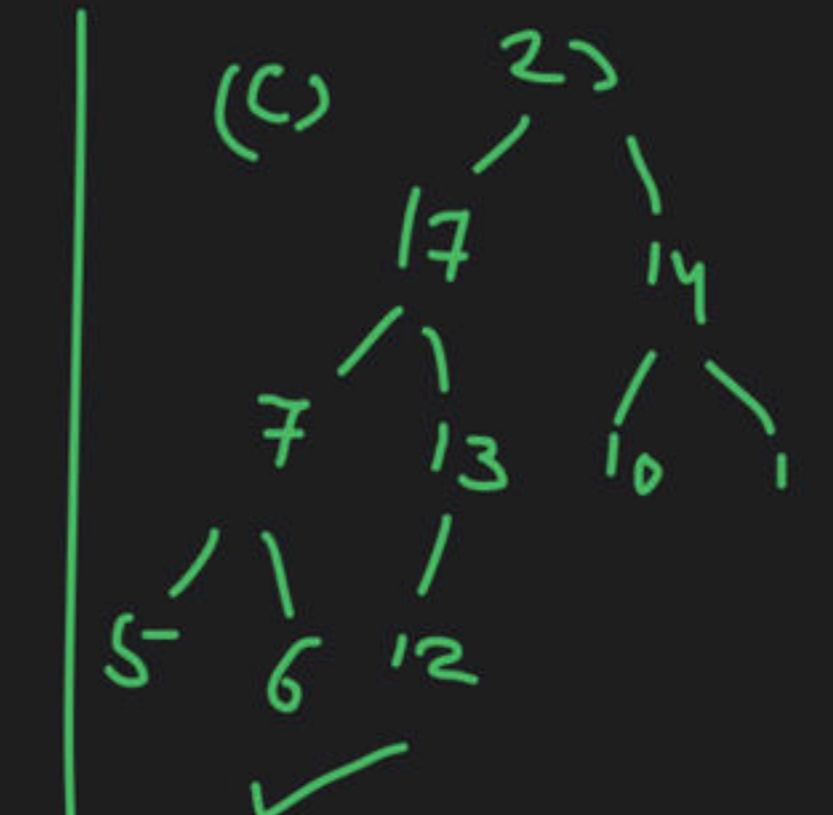
(A)



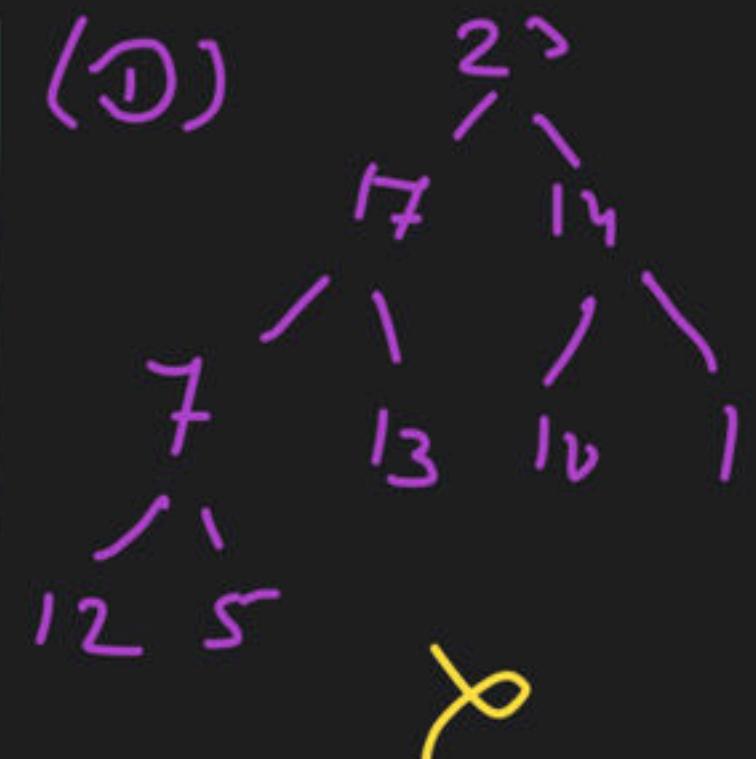
(B)



(C)



(D)

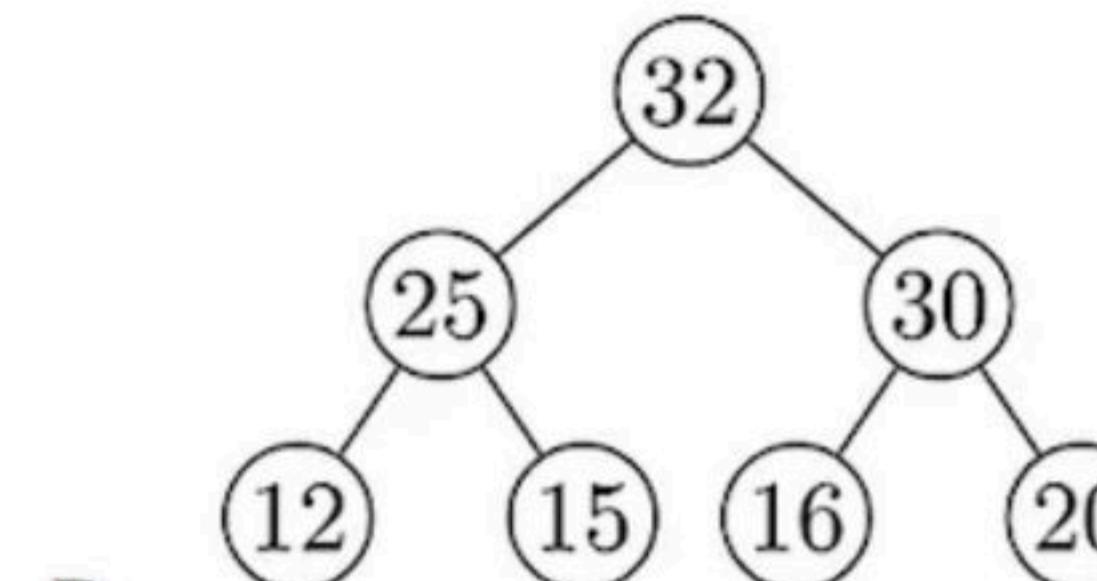
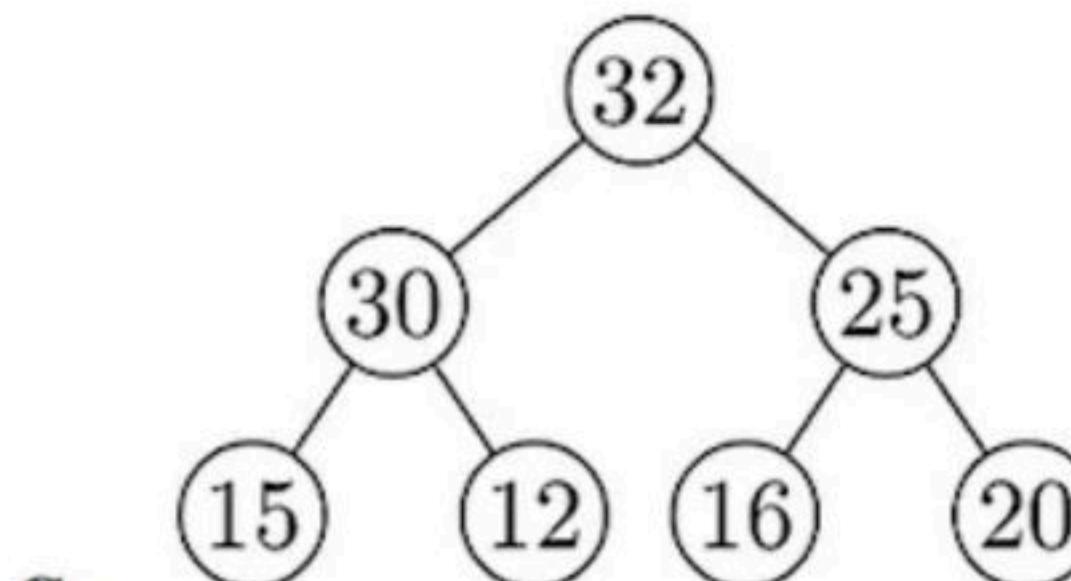
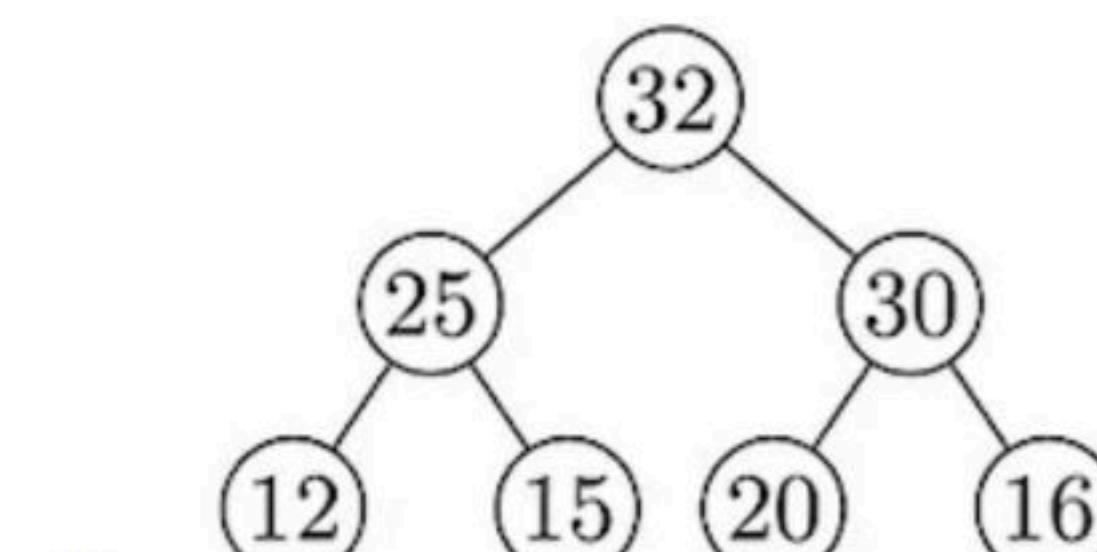
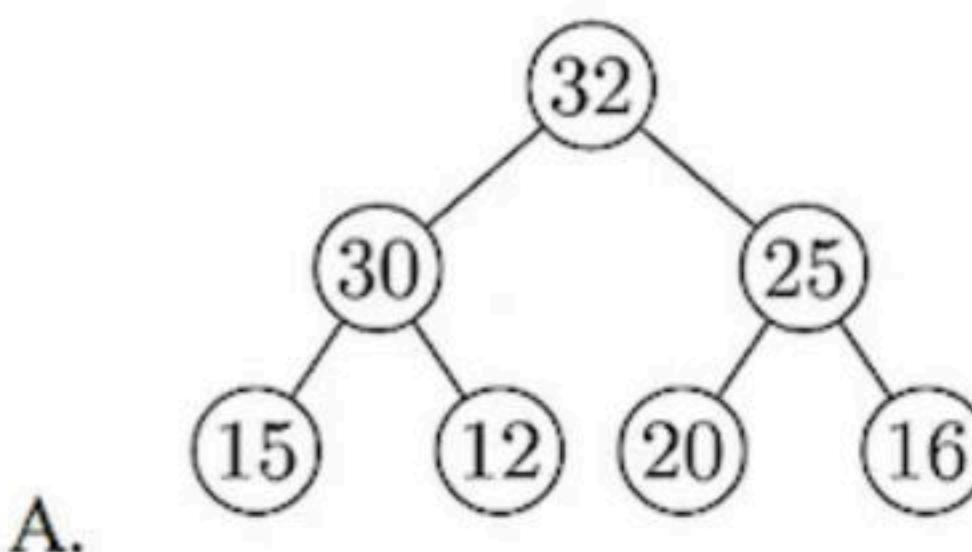


Which of the following sequences of array elements forms a heap?

- A. {23, 17, 14, 6, 13, 10, 1, 12, 7, 5}
- C. {23, 17, 14, 7, 13, 10, 1, 5, 6, 12}

- B. {23, 17, 14, 6, 13, 10, 1, 5, 7, 12}
- D. {23, 17, 14, 7, 13, 10, 1, 12, 5, 7}

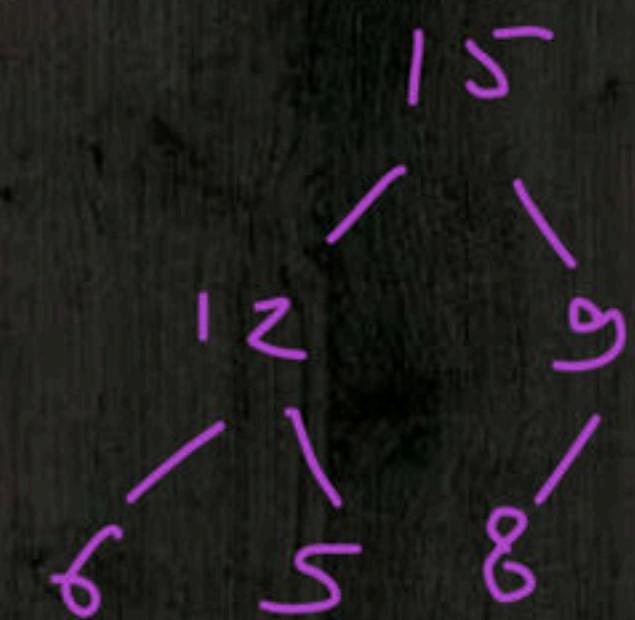
The elements 32, 15, 20, 30, 12, 25, 16, are inserted one by one in the given order into a maxHeap. The resultant maxHeap is



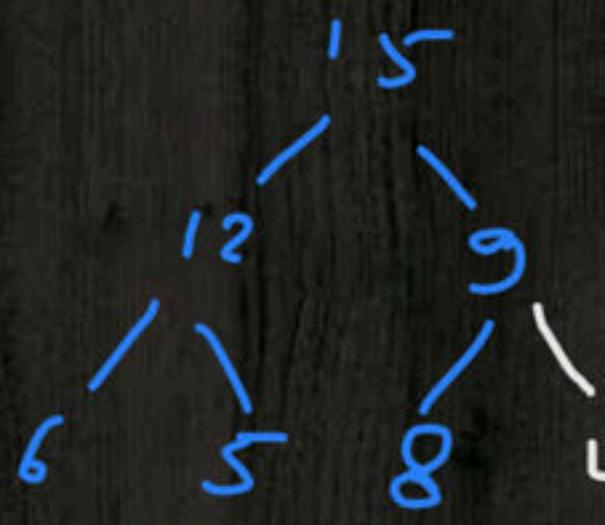
Insertion in Heap

→ first add new element based on Complete BT,
then adjust tree for heap property.

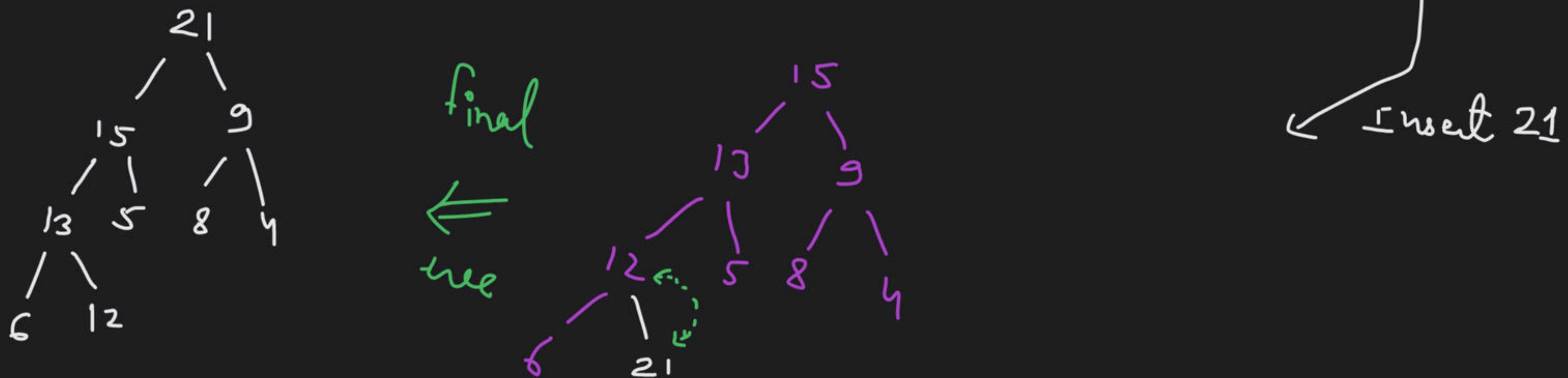
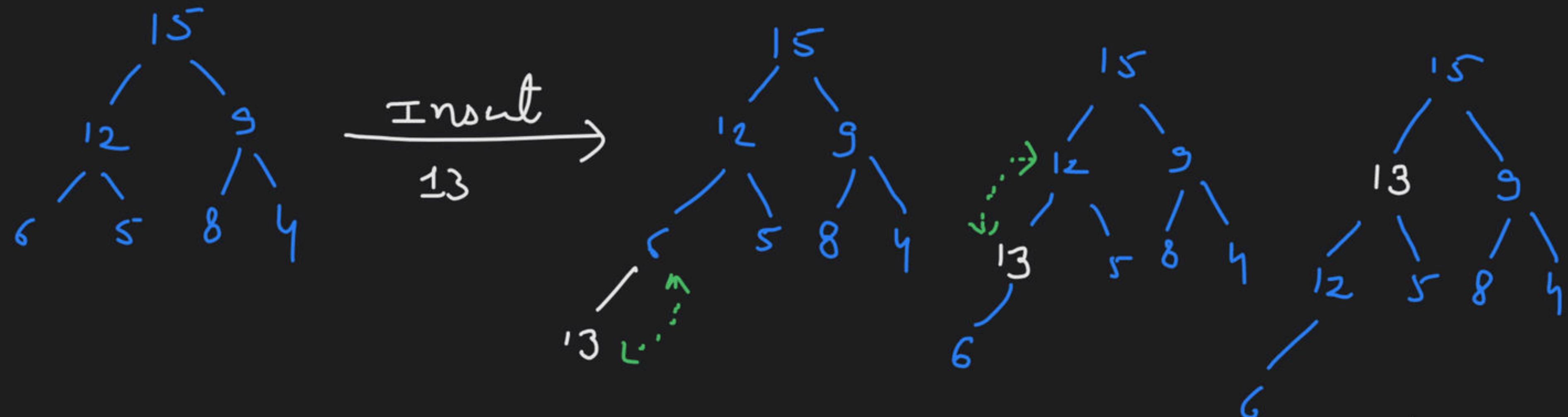
max heap :-



insert => 4



no adjustment
needed

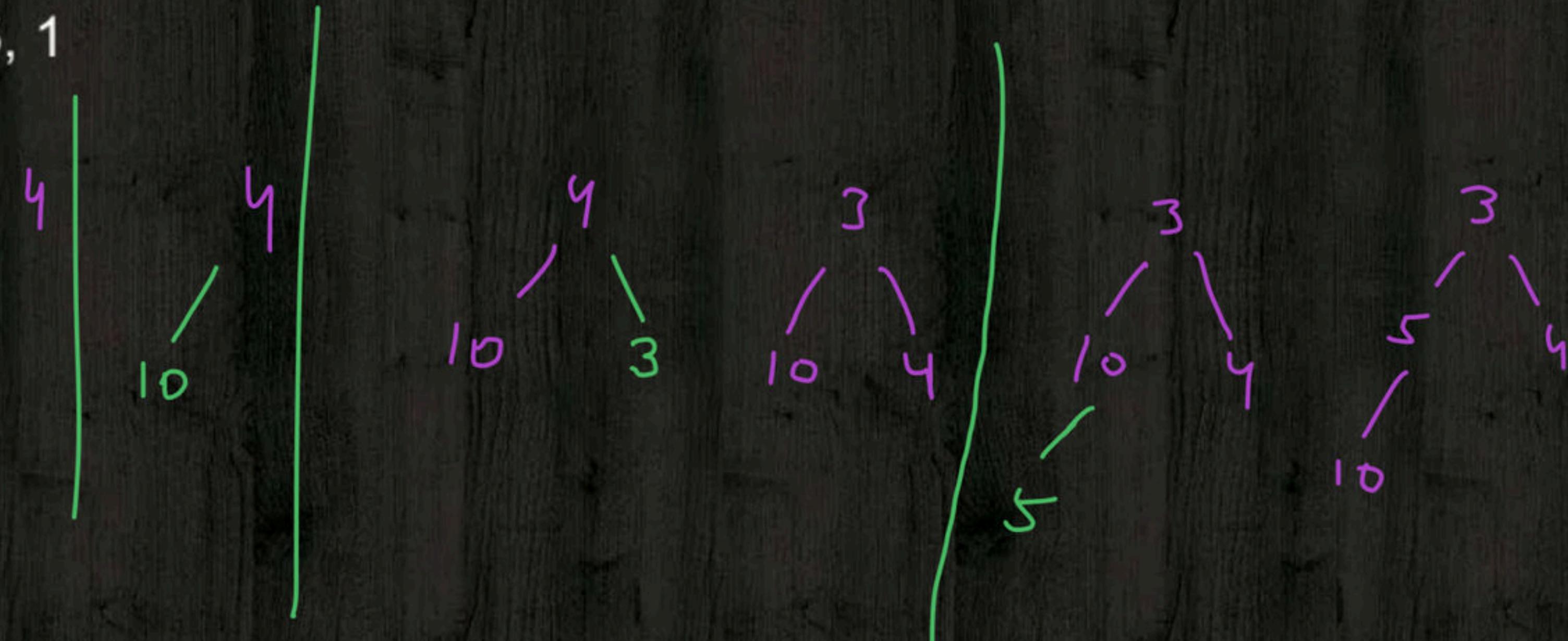


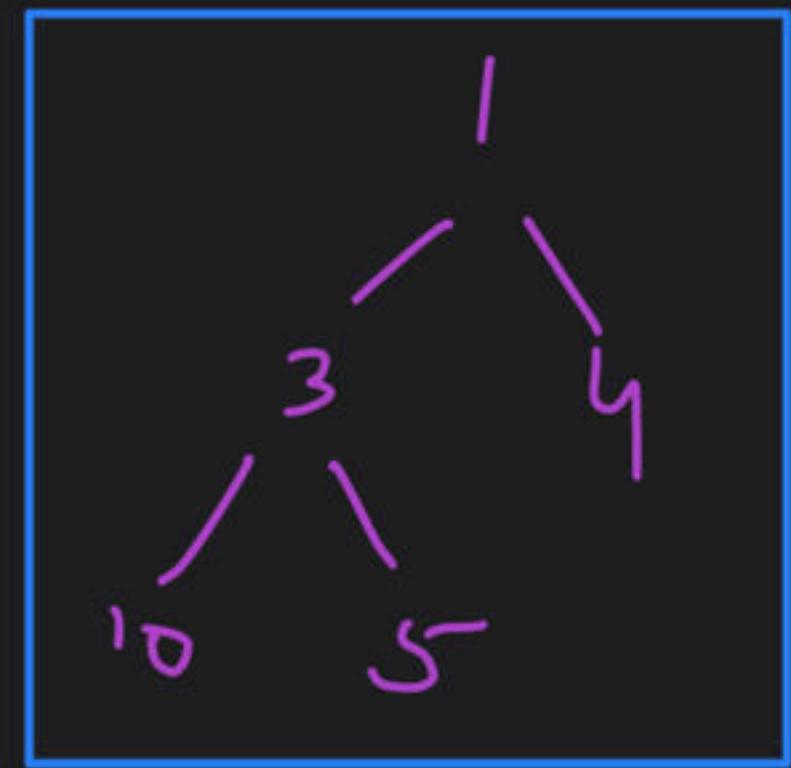
Run time complexity
for insertion in heap $\Rightarrow O(\log_2 n)$

Building a Heap

Min Heap:

4, 10, 3, 5, 1



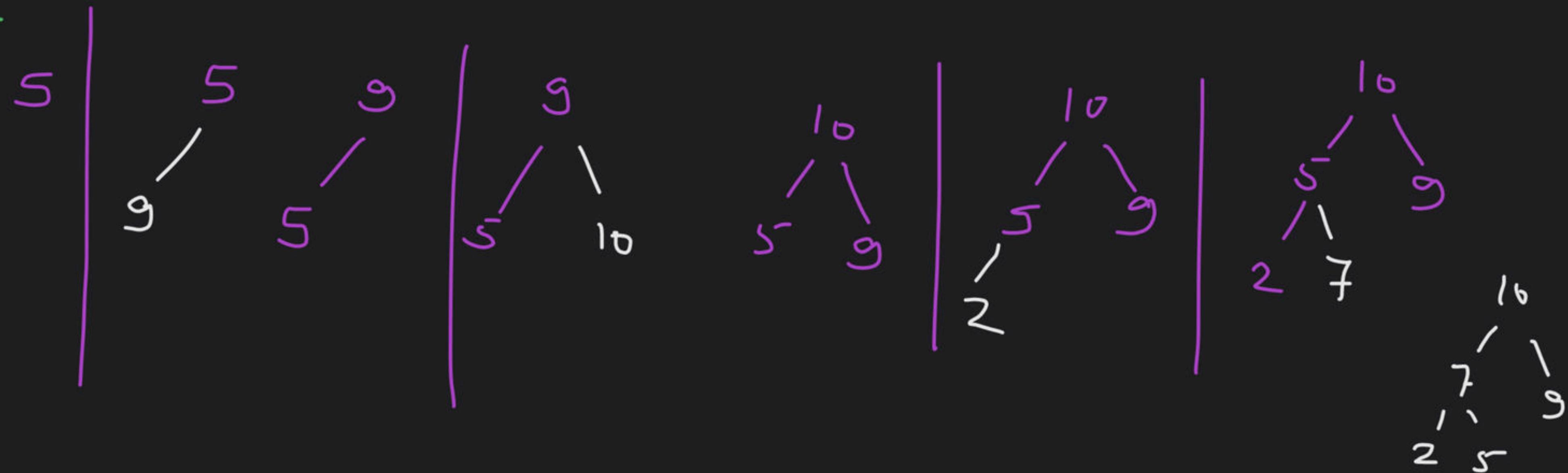


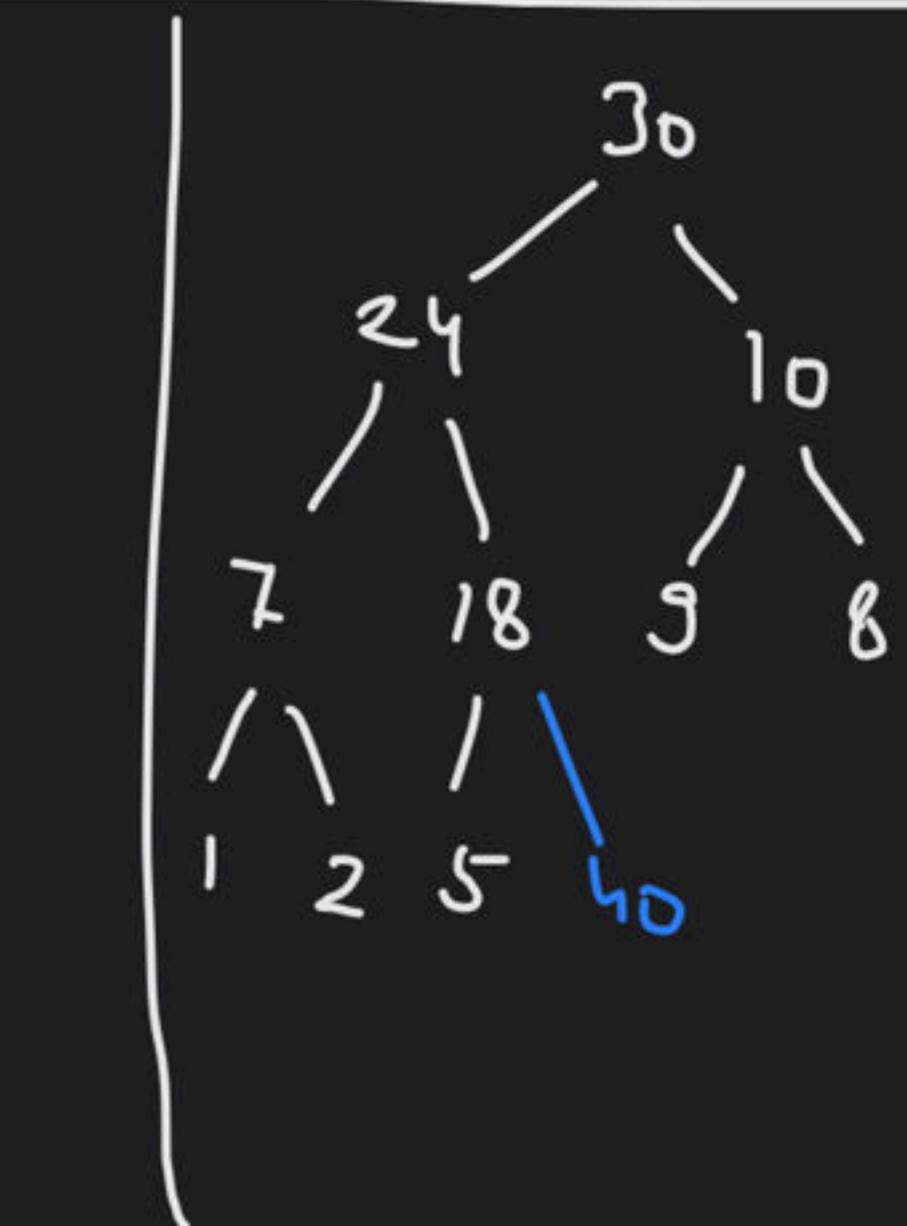
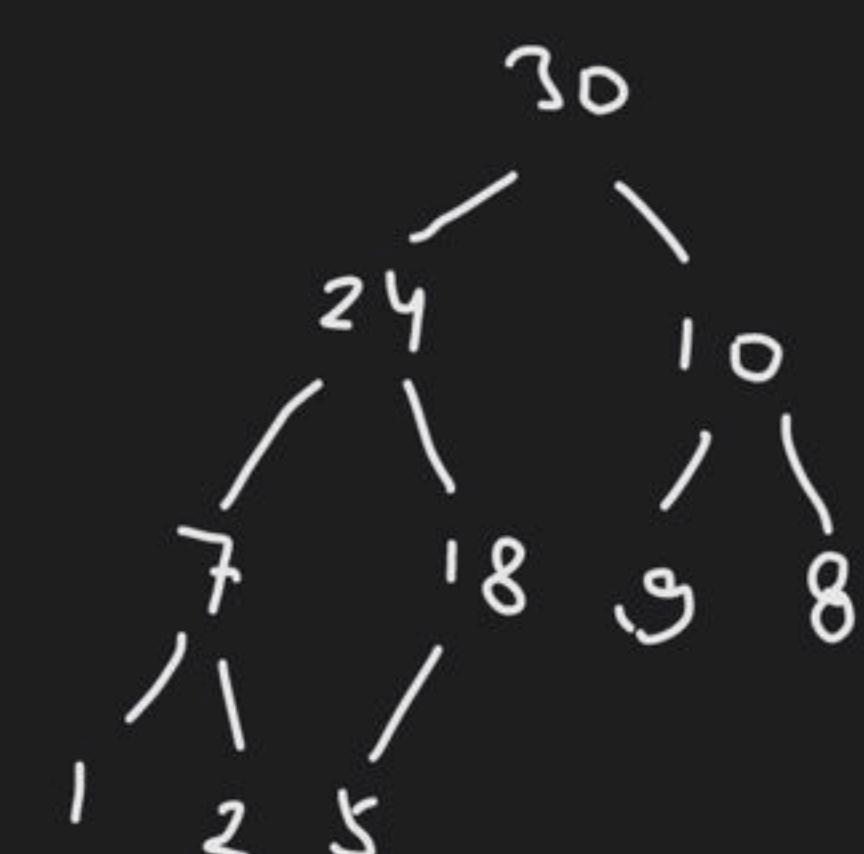
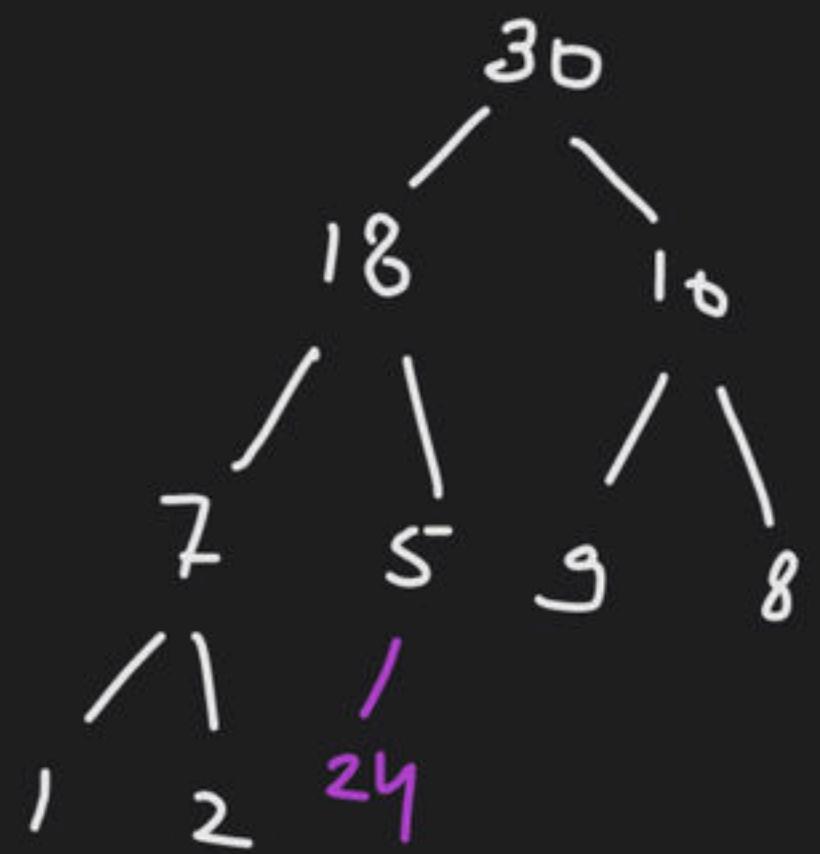
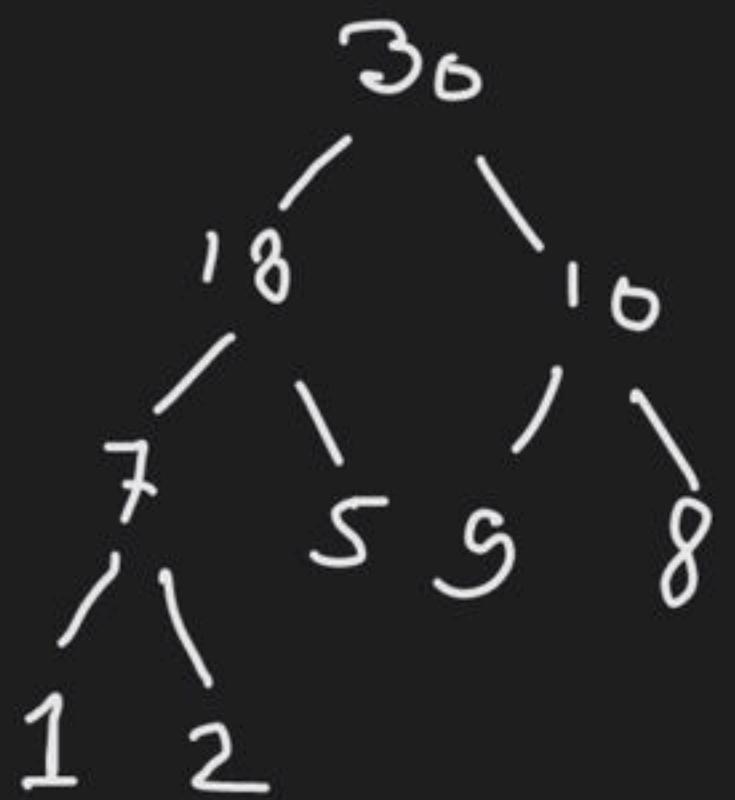
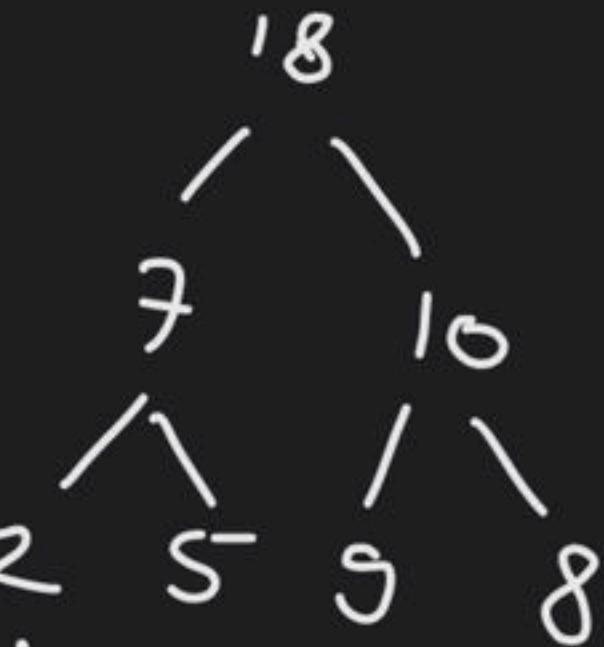
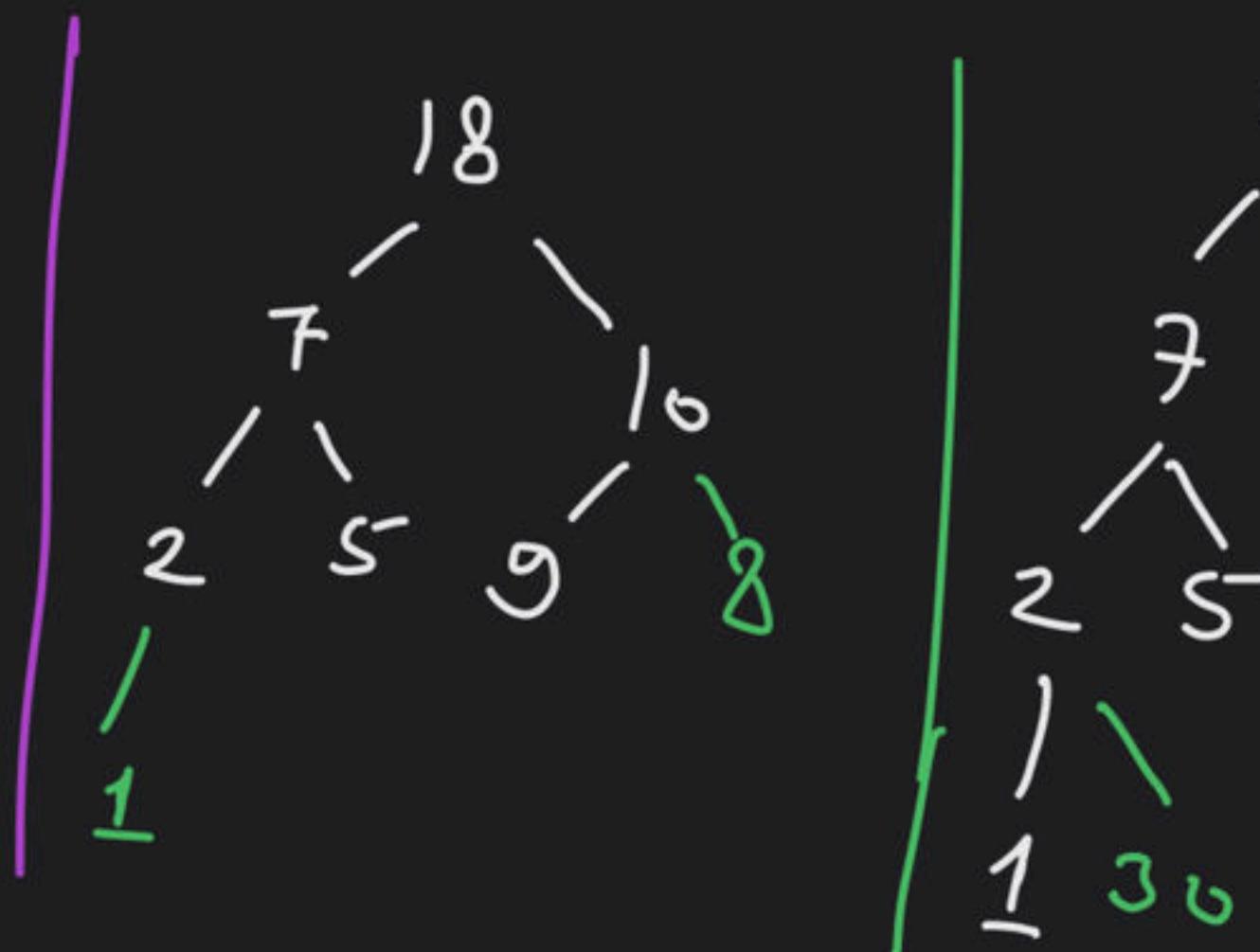
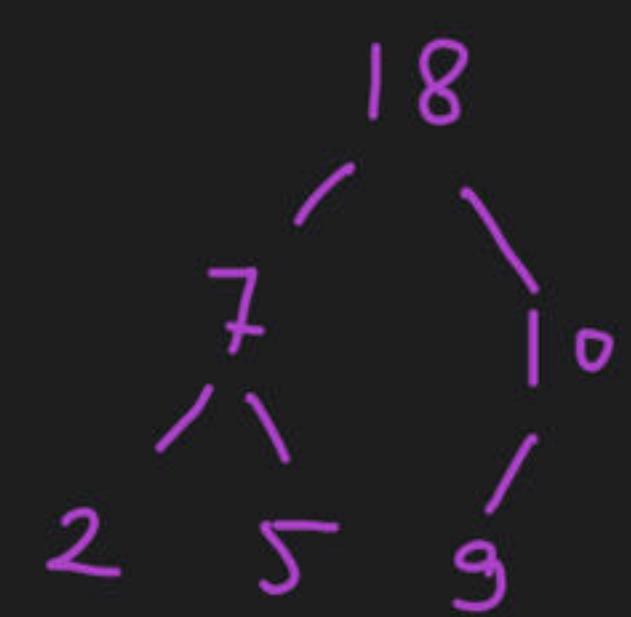
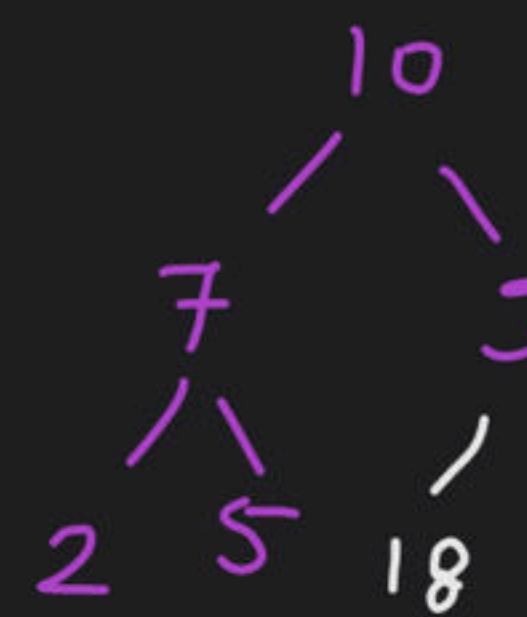
R.T. Complexity
⇒ Building heap (one-by-one insertion) ⇒ $n \log_2 n$

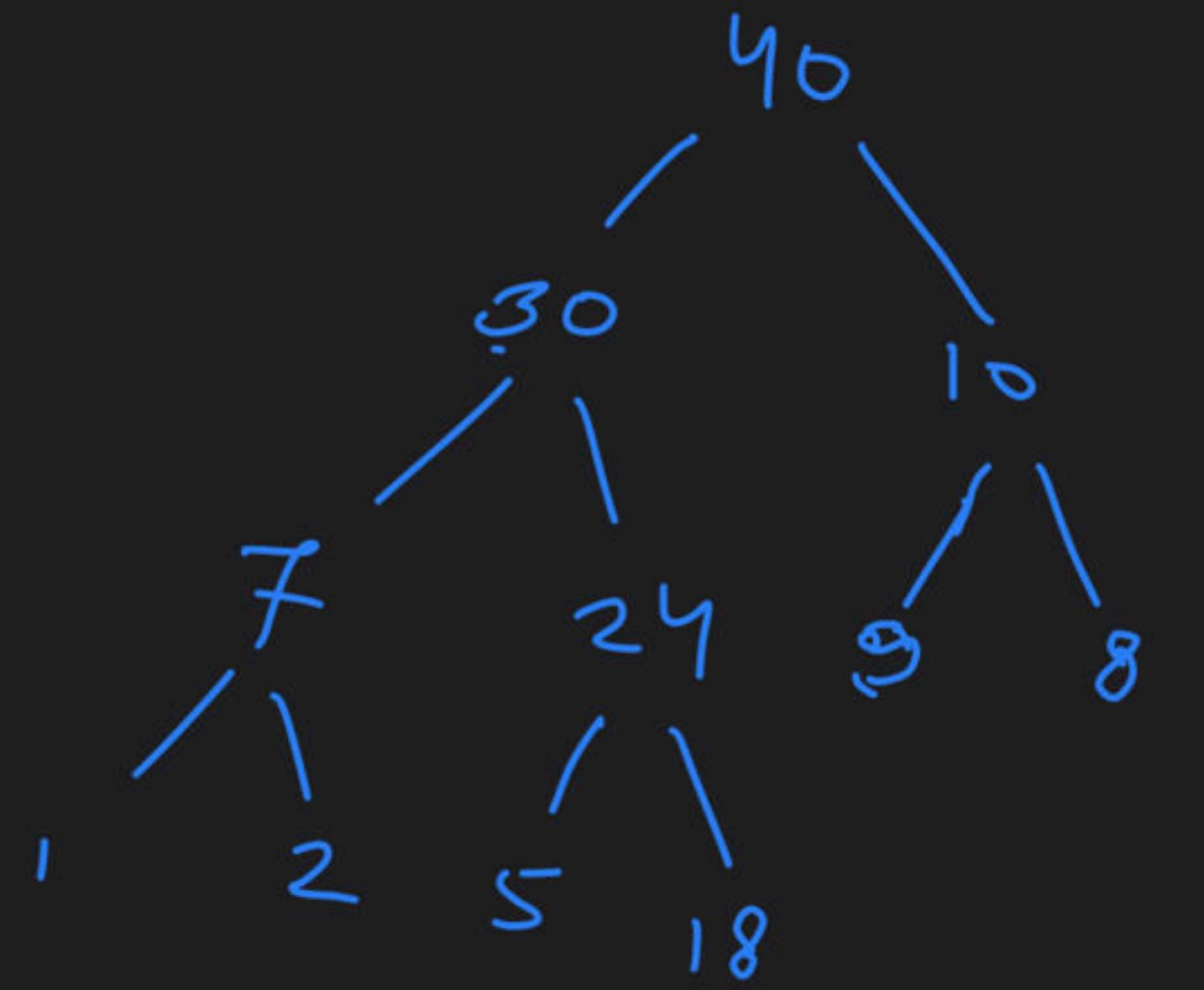
ques) construct max heaps. through one-by-one insertion?

5, 9, 10, 2, 7, 18, 8, 1, 30, 24, 40

Soln







Building a Heap

Max Heap:

1, 3, 5, 4, 6, 13, 10, 9, 8, 15, 17

Building a Heap: Runtime

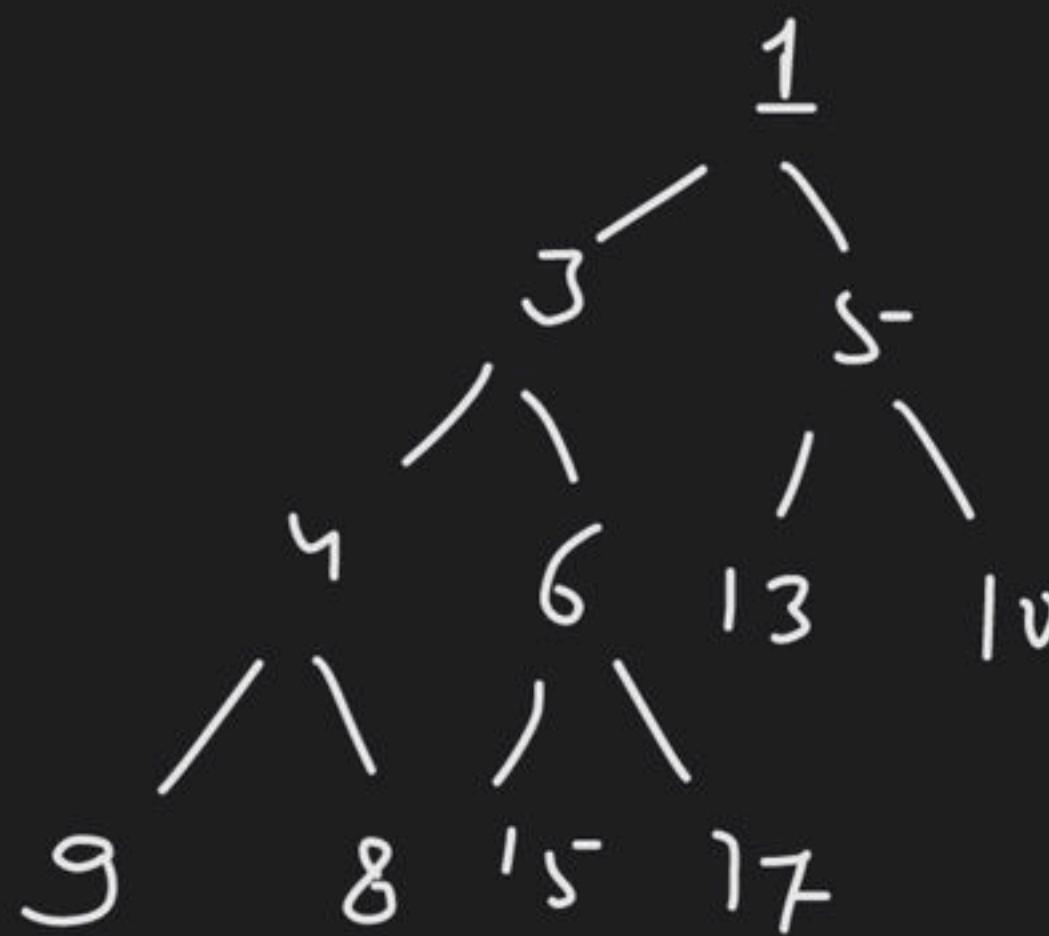
Building a Heap: Heapify

Max Heap:

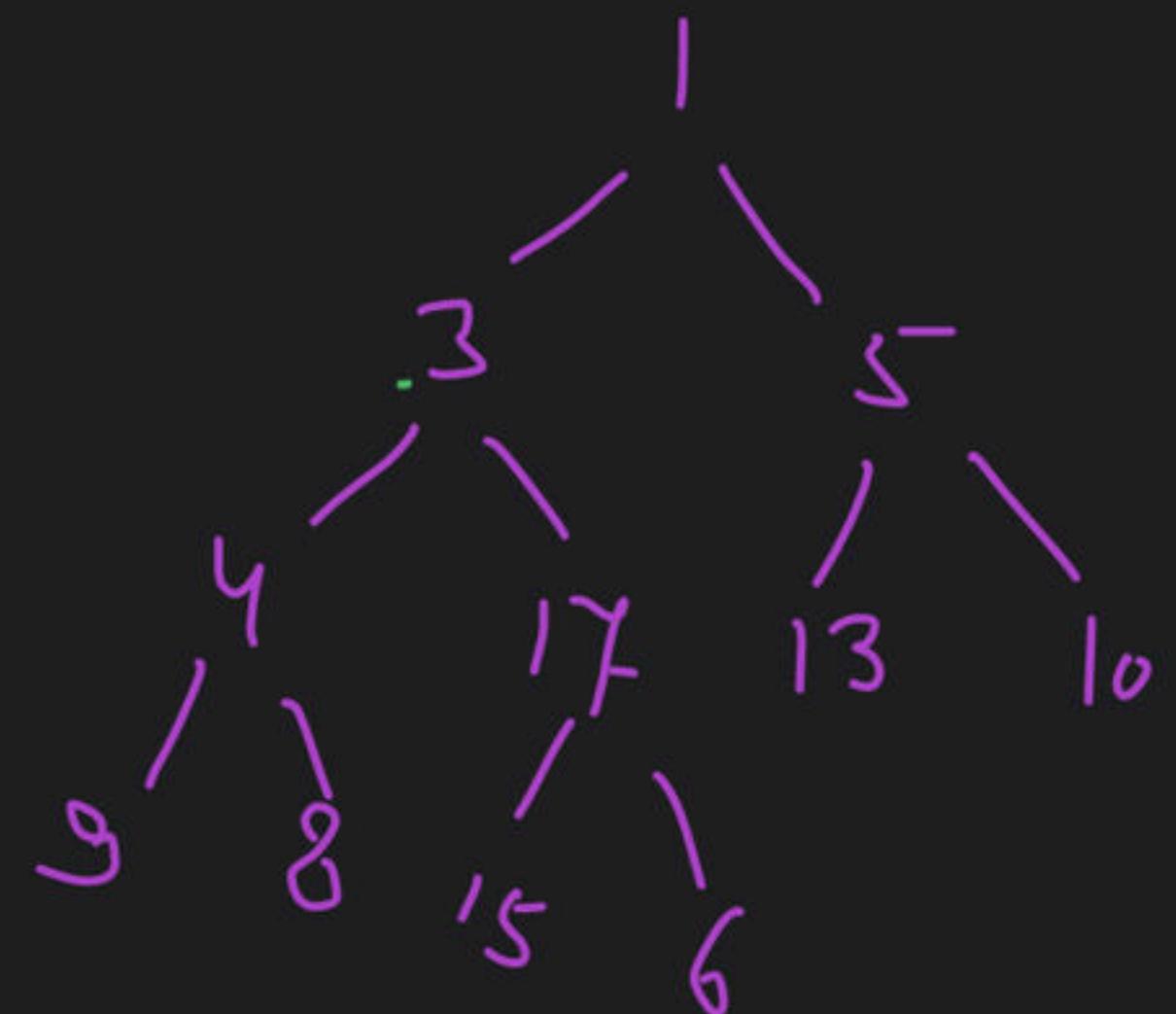
1, 3, 5, 4, 6, 13, 10, 9, 8, 15, 17

- ⇒ Construct complete BT using given keys.
- ⇒ from last to first internal node, heapify

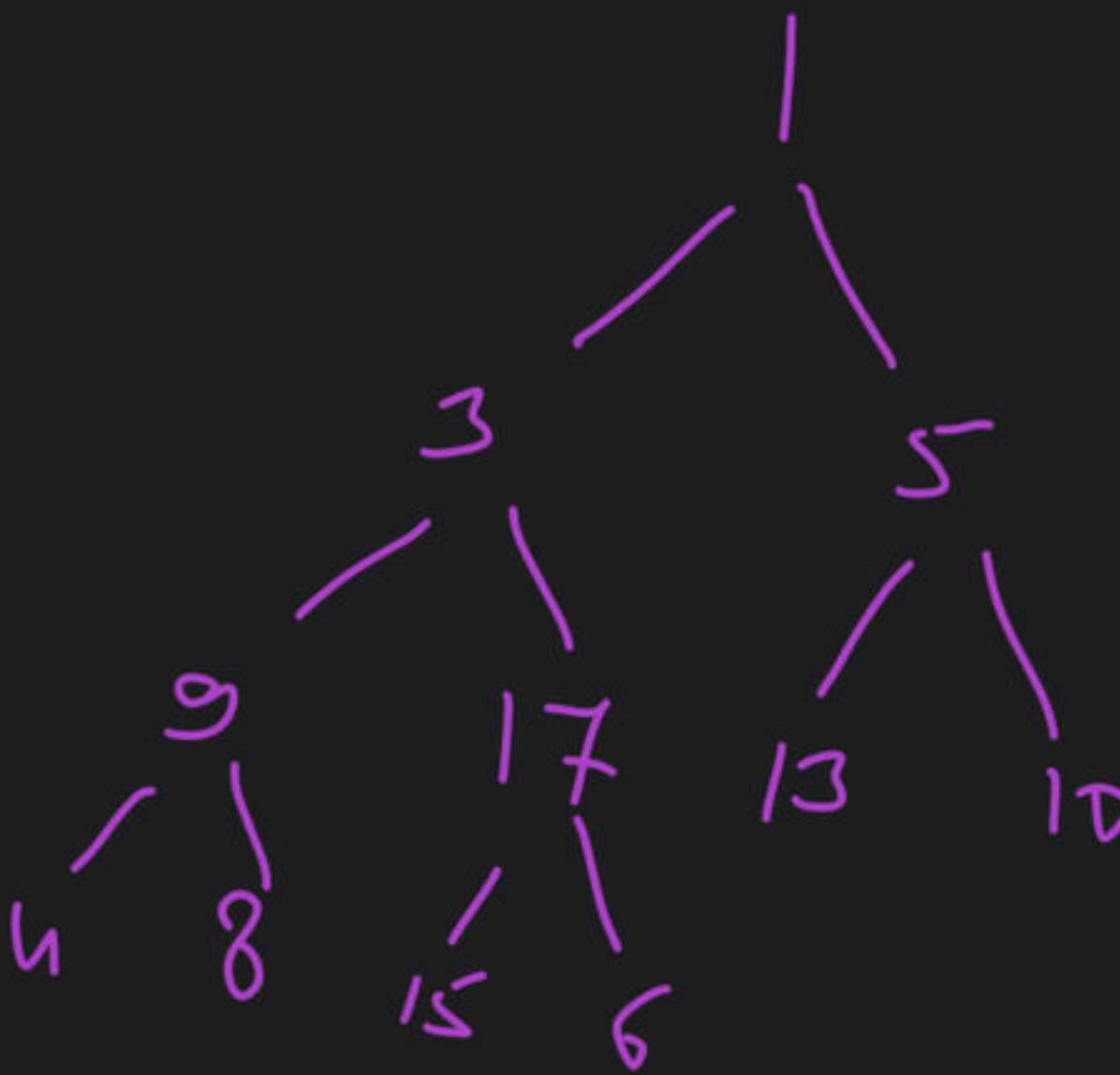
C.B.T



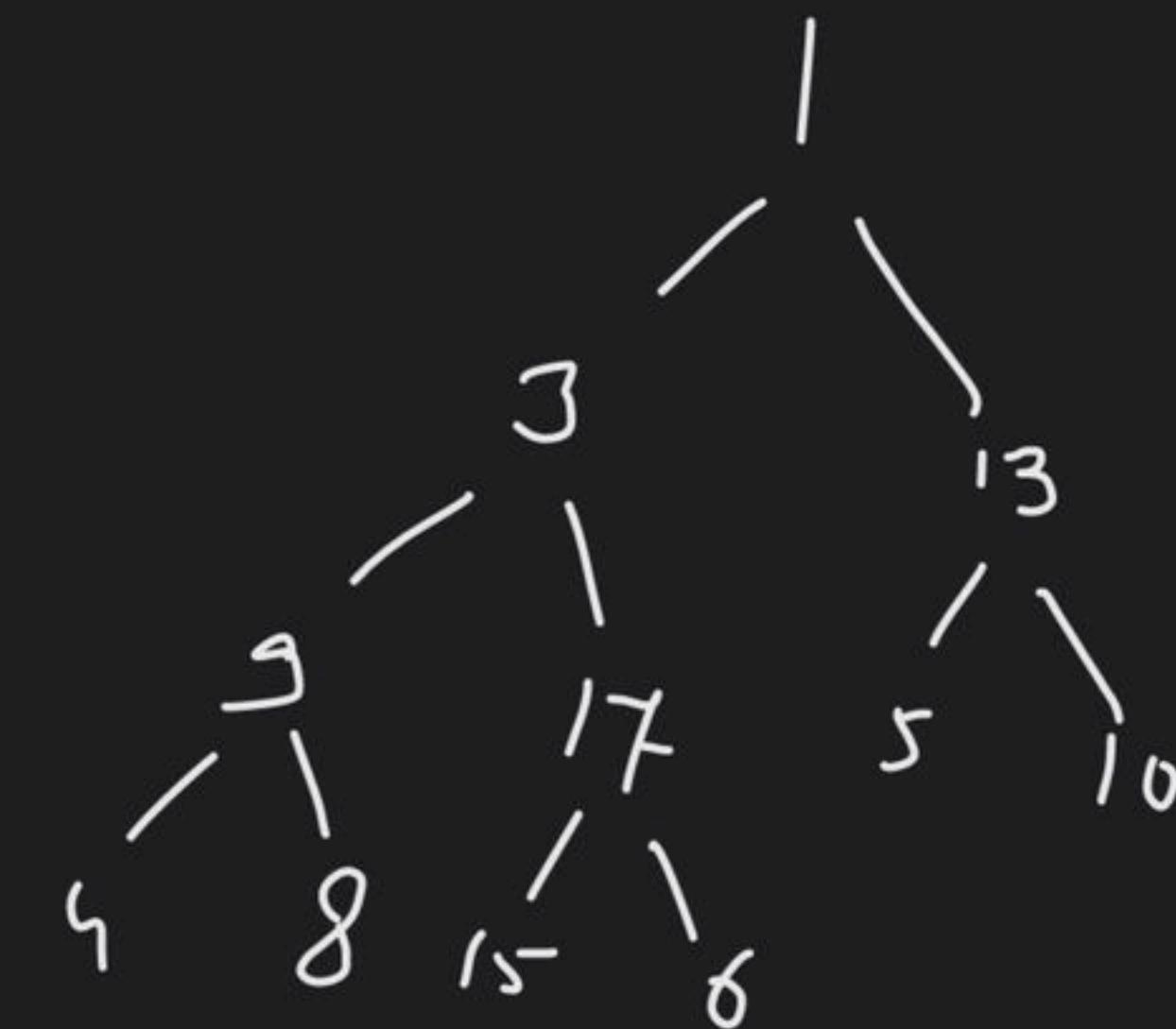
Reapify at 6:- Adjust 6, such that
the subtree starting from 6 as root
becomes max heap.



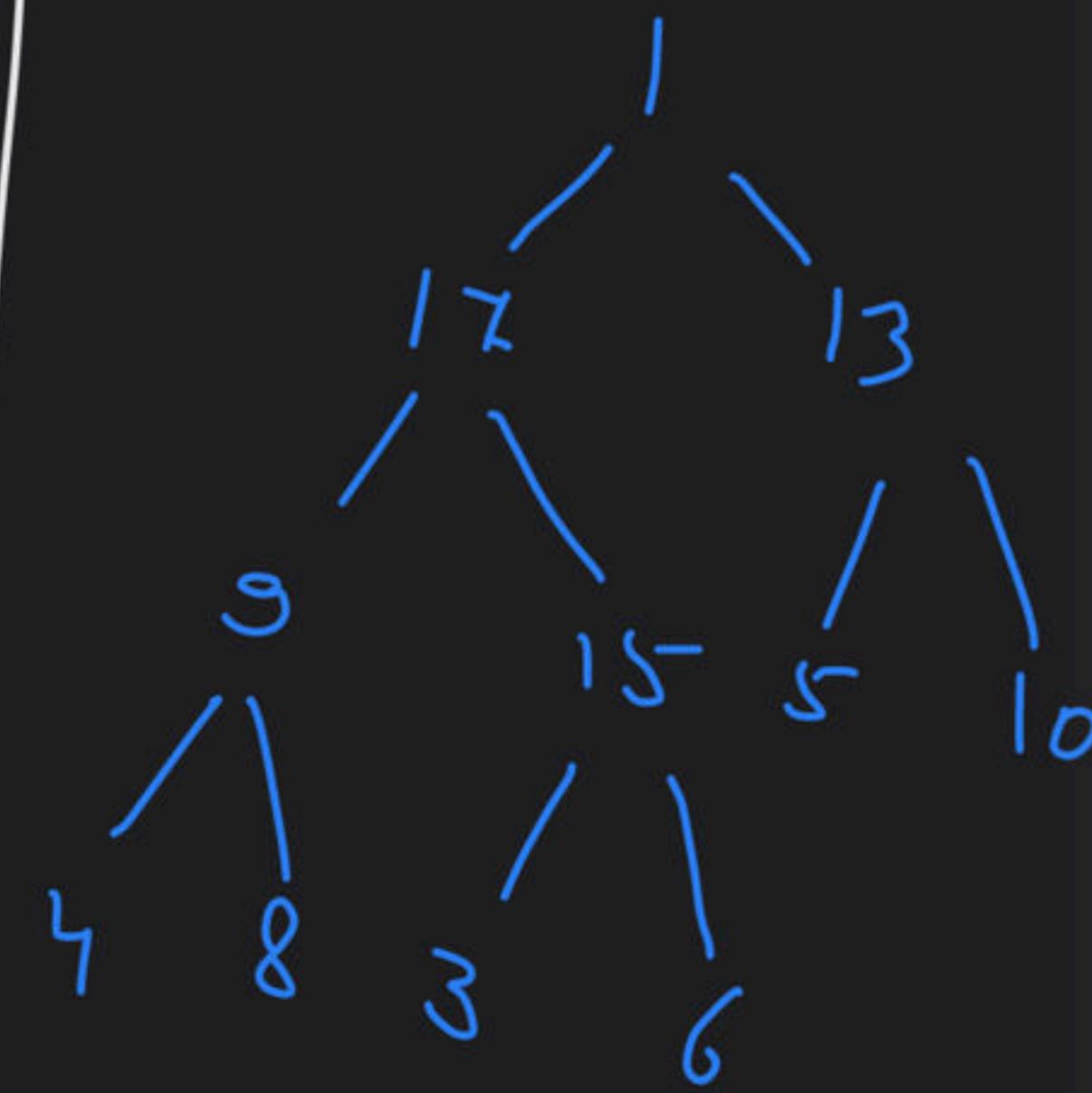
Heapify at 4:-



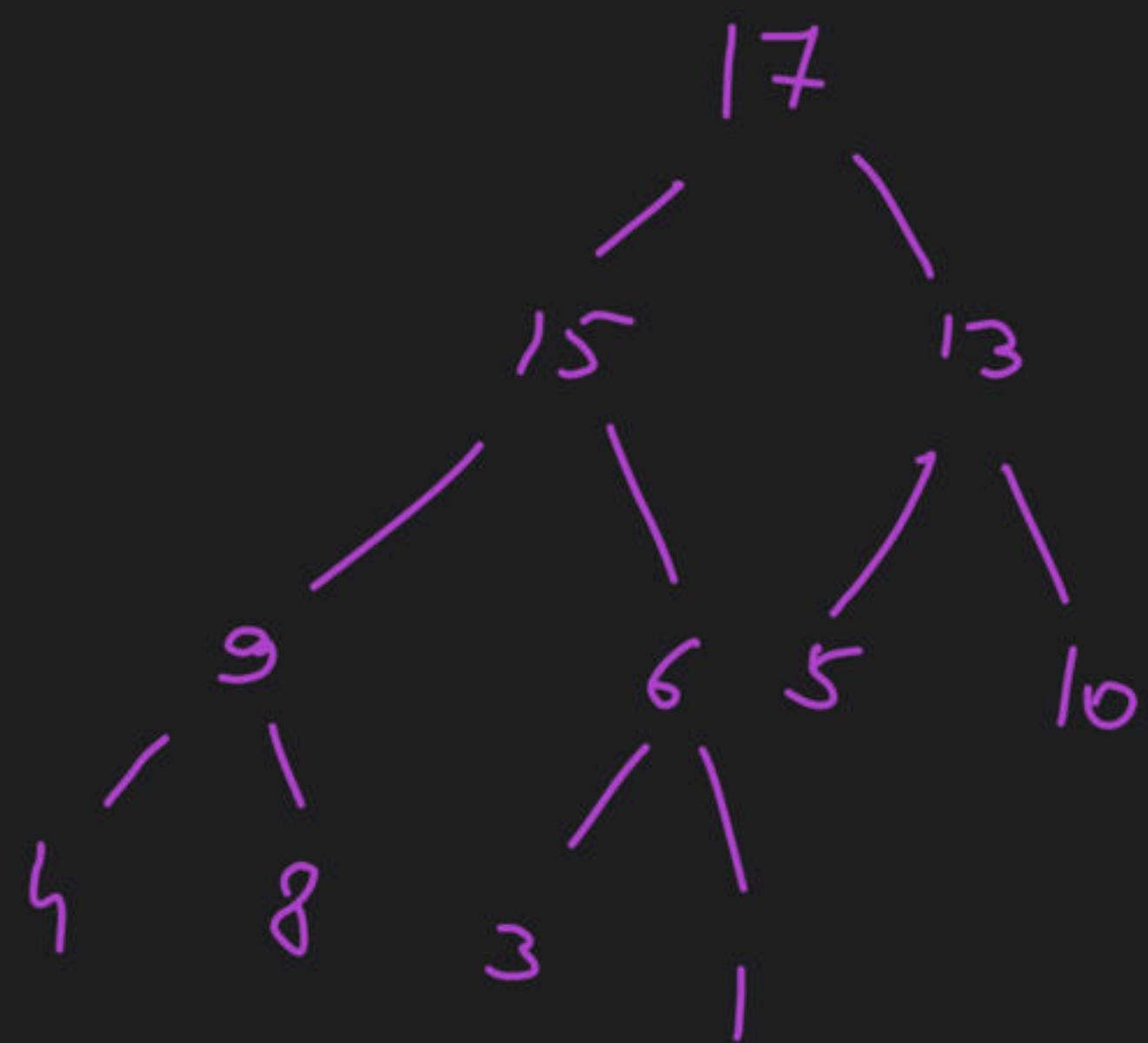
Heapify at 5:-



Heapify at 3:-



Heapify at 1:



Building a Heap: Heapify Runtime

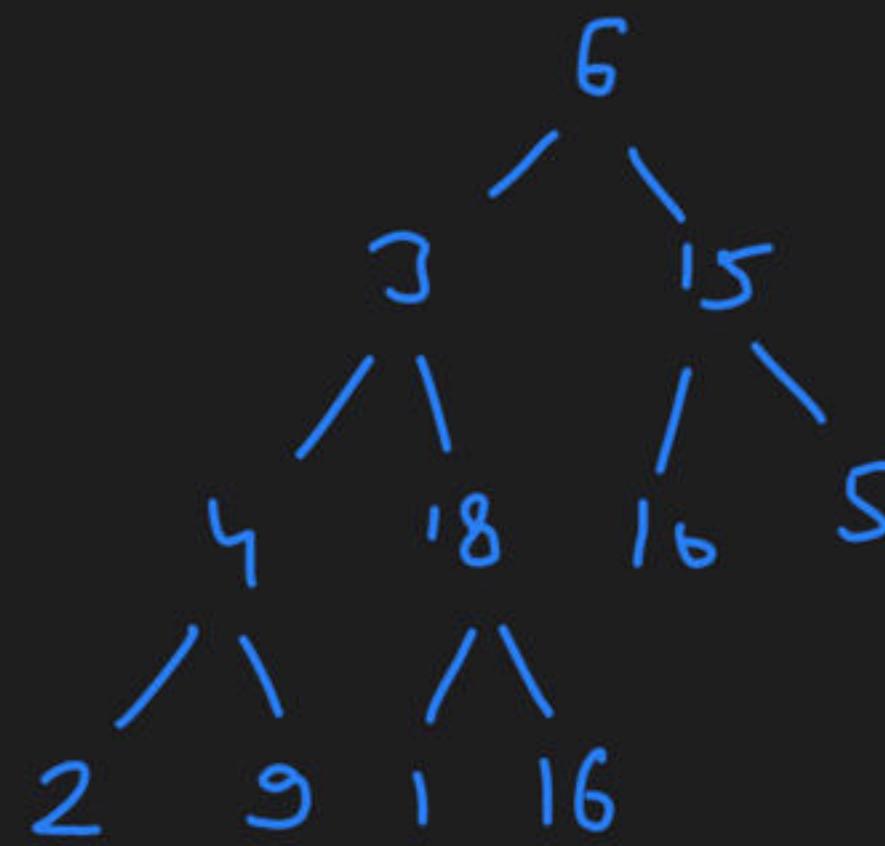
$$\Rightarrow \Theta(n)$$

with
 n element

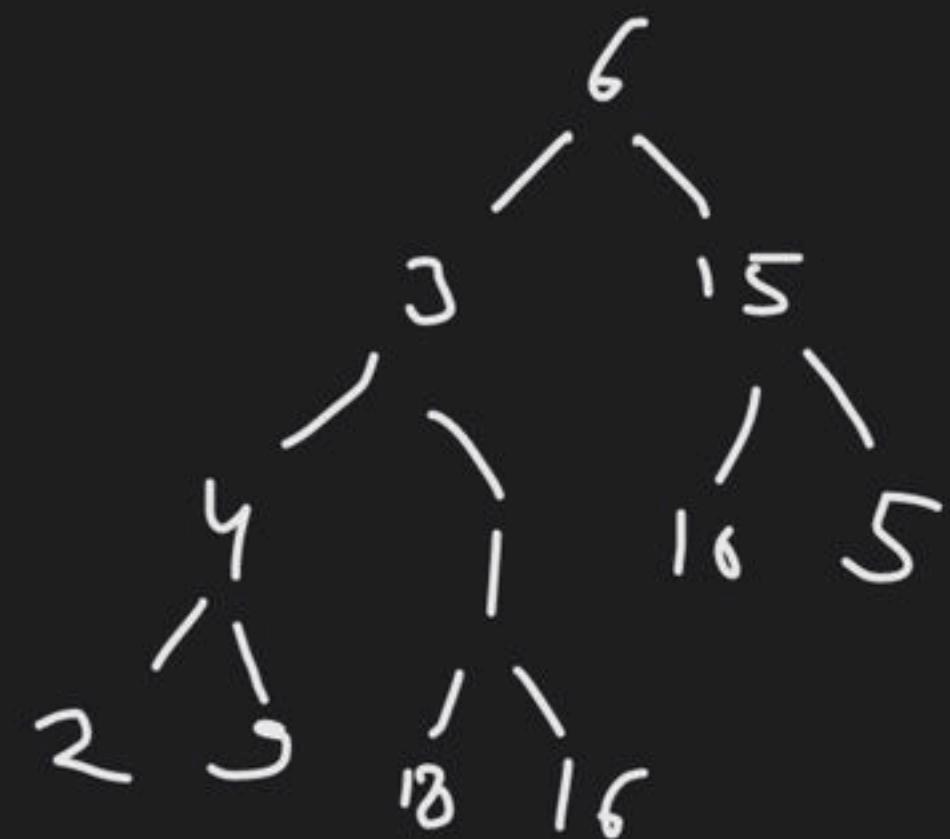
Ans) Build min-heap using heapify method :-

6, 3, 15, 4, 18, 10, 5, 2, 9, 1, 16

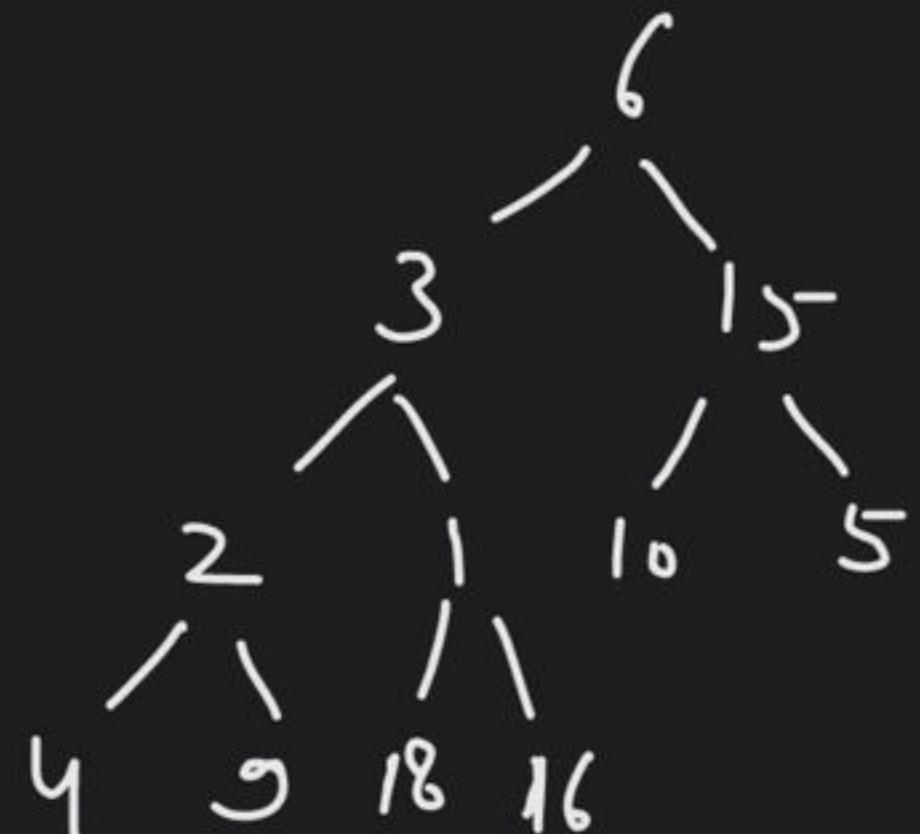
C.B.T.



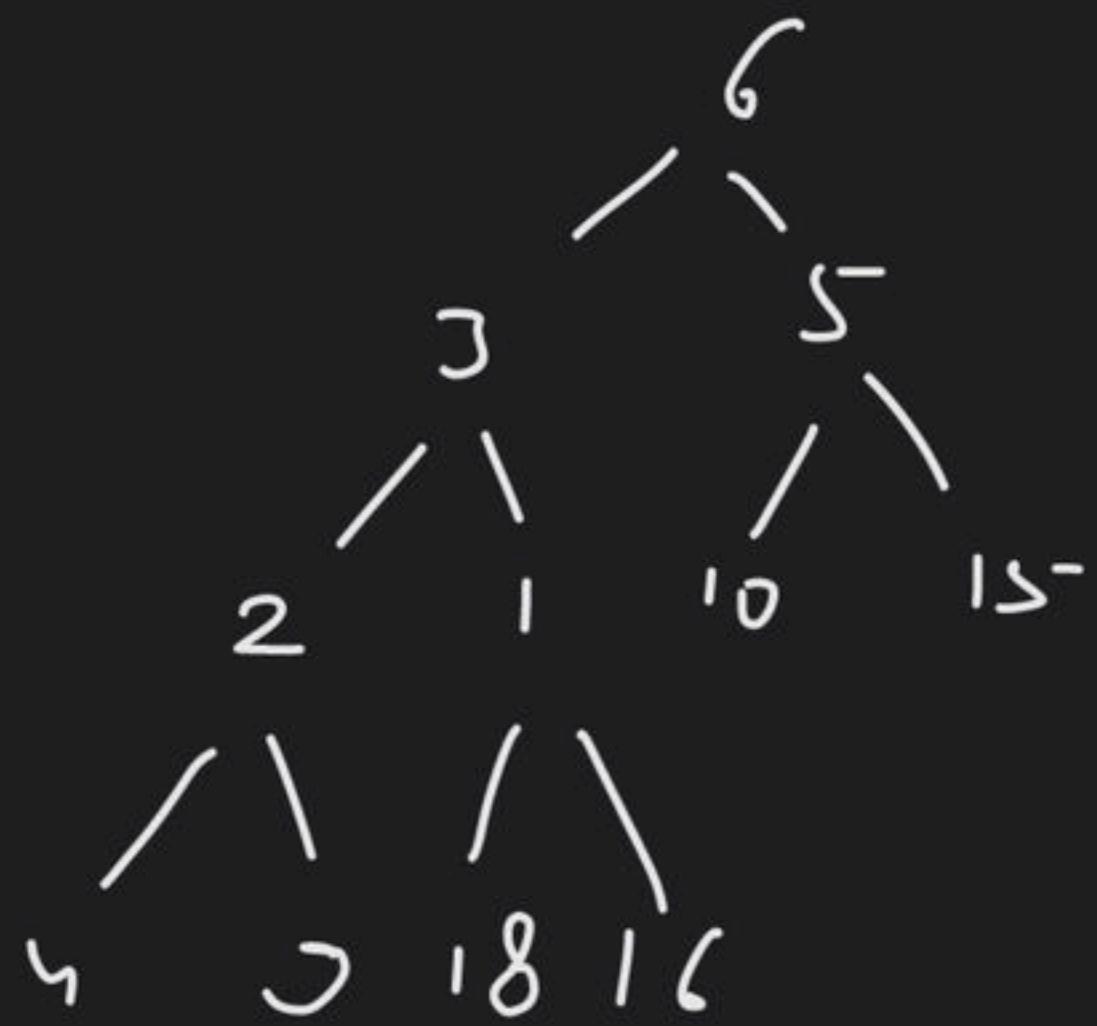
Heapify 18 :-



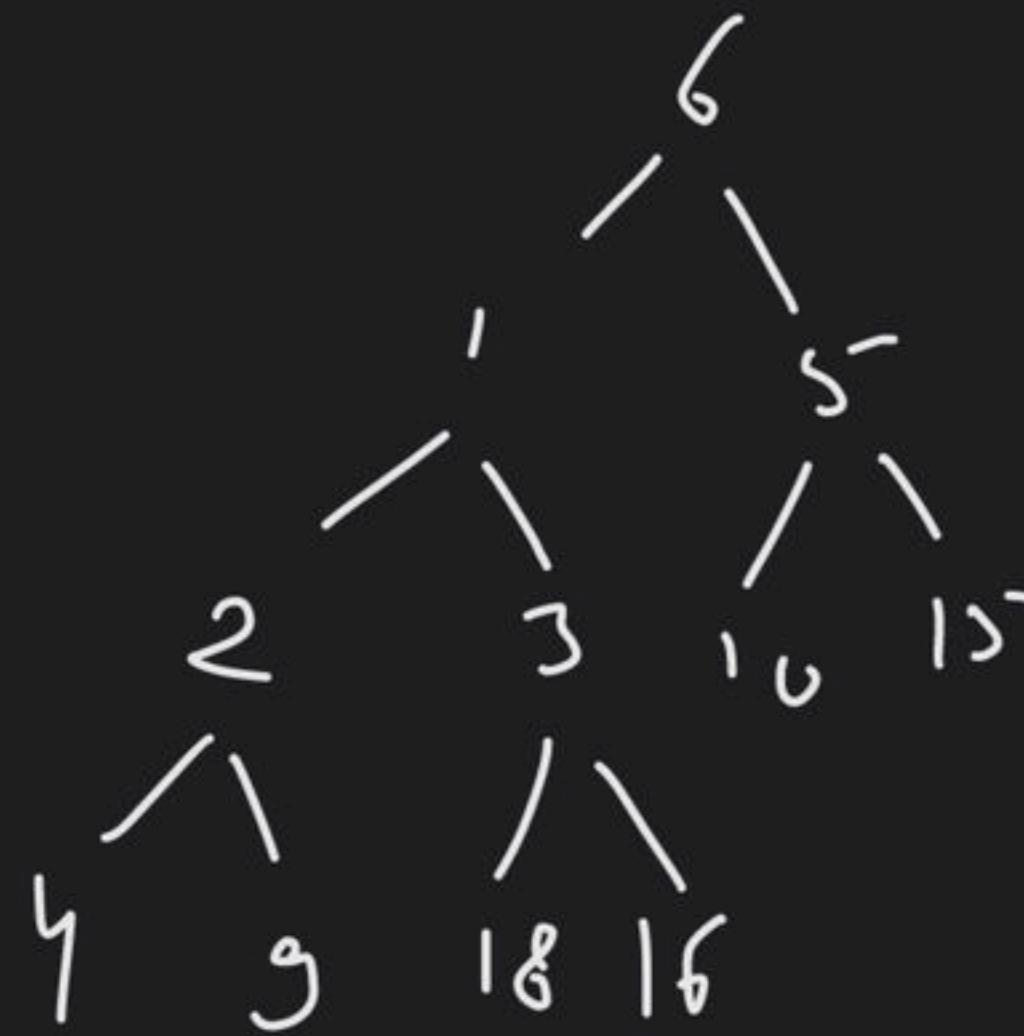
Heapify 4 :-



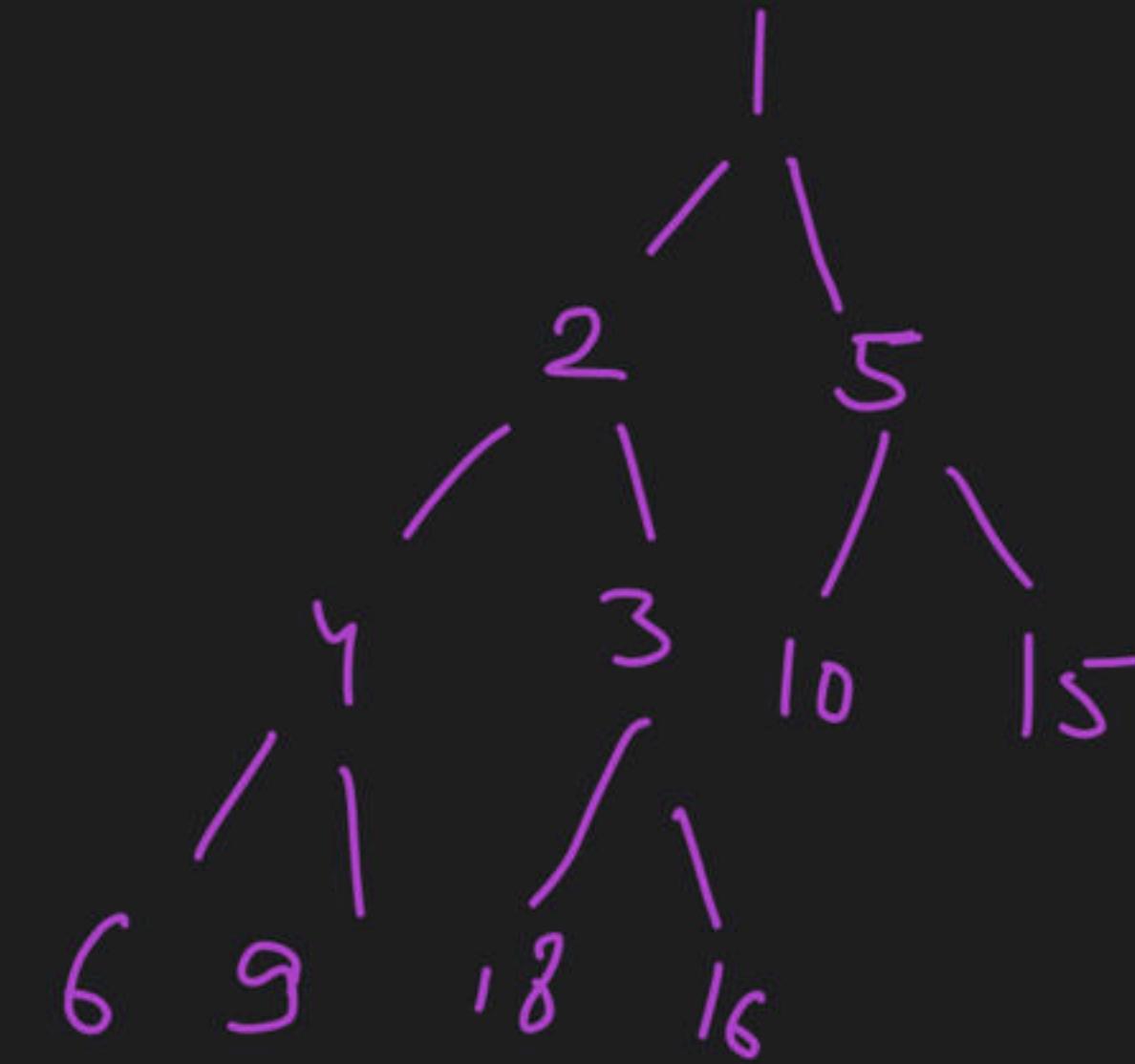
Heapify at 15:-



Heapify at 3:-



Heapify at 6:-



Ques)

Build min heap , using one - by - one , and calculate no. of swaps (interchanges) of elements ?

1 , 2 , 3 , 4 , 5

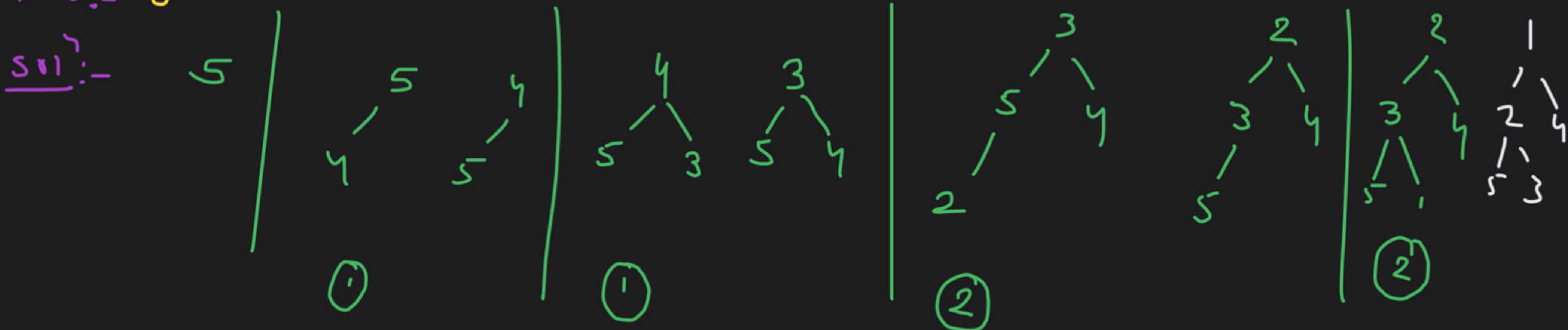
Ans:- zero

Sol :- If elements are given in ascending order then 0 swaps required to build min-heap .

Ques) Build min-heaps , using one-by-one insert. Calculate total no. of swaps ?

5, 4, 3, 2, 1

Ans:- 6



Ques) Build min heap , Using one - by - one insertn .
no of swaps ?

15, 14, -- , 1

Ans :-

Deletion In Heap

Number of Heaps Using N keys

Number of Heaps Using N keys

$$T(N) = (N - 1)Ck * T(k) * T(N - k - 1)$$

Where k = number of nodes on left subtree

Question

The number of possible min-heaps containing each value from {1,2,3,4,5,6,7} exactly once is ?

Happy Learning