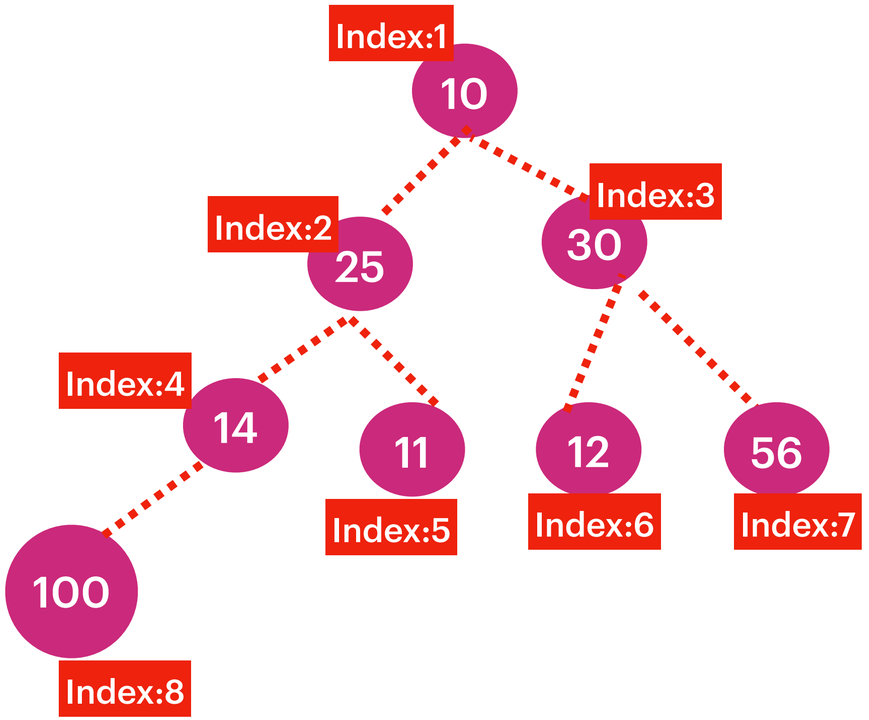
## Complete Binary Tree




## Transform Complete Binary Tree to the Hash/Array

How to you find parent ?

How do you find left & right child ?

How do you know that current element is a leaf Node ?

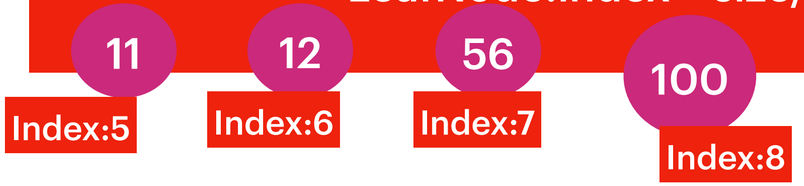
Left Child = index * 2
Right Child = index * 2 + 1

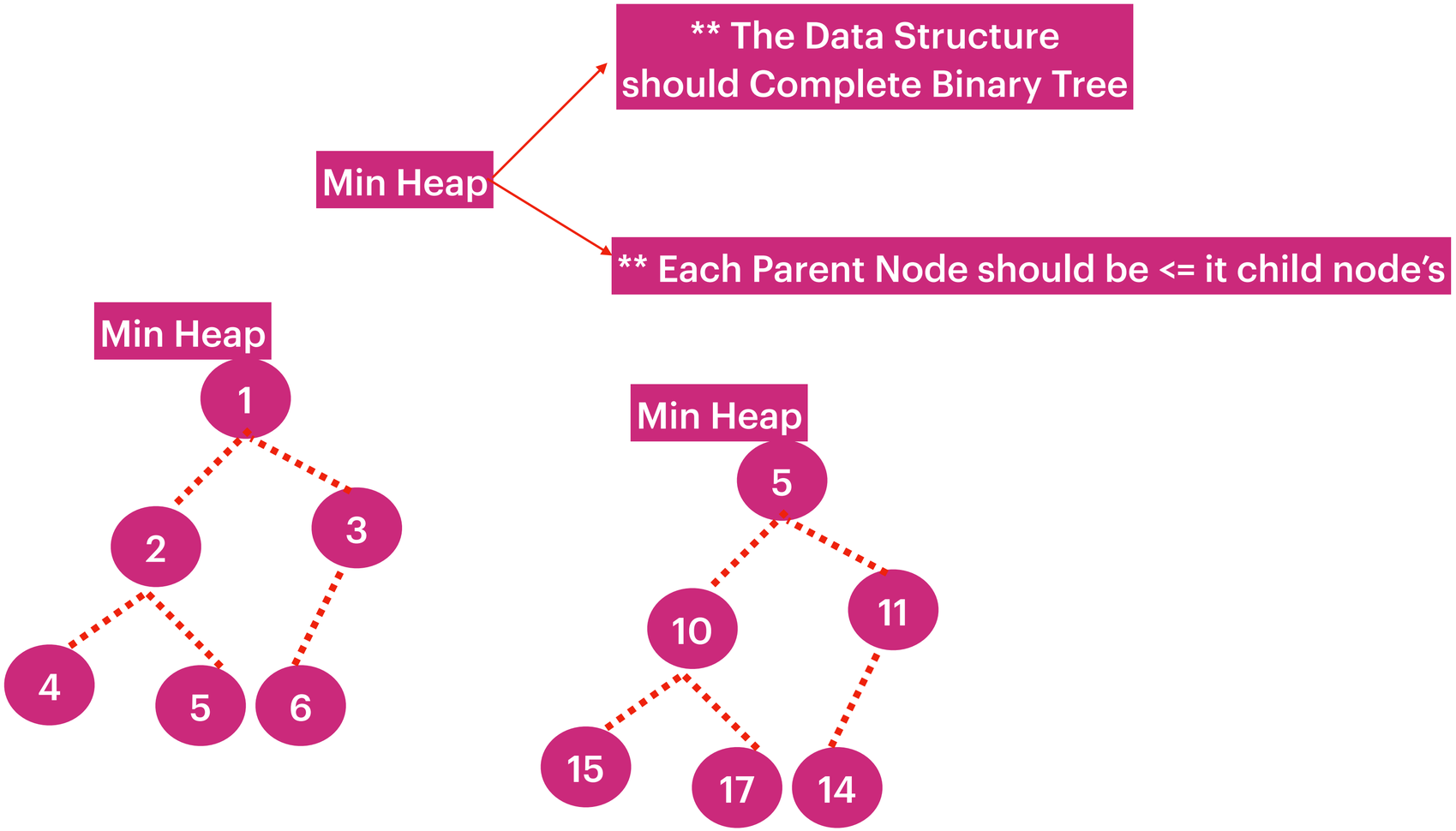
Ex: Left Child of Element 30 = 2 * index:3 = index:6 = 12
Right Child of Element 30 = 2 * index:3 + 1 = index:7= 56

Parent = index/2;

Ex: Parent Of 25 = 2/2 = index:1 = 10

How do you know that current element is a leaf Node ?
LeafNode:index > size/2




** The Data Structure
should Complete Binary Tree
Min Heap
** Each Parent Node should be <= it child node's
Min Heap
1
2
3
4
5
6
Min Heap
5
10
11
15
17
14

# Insert Operation On MinHeap

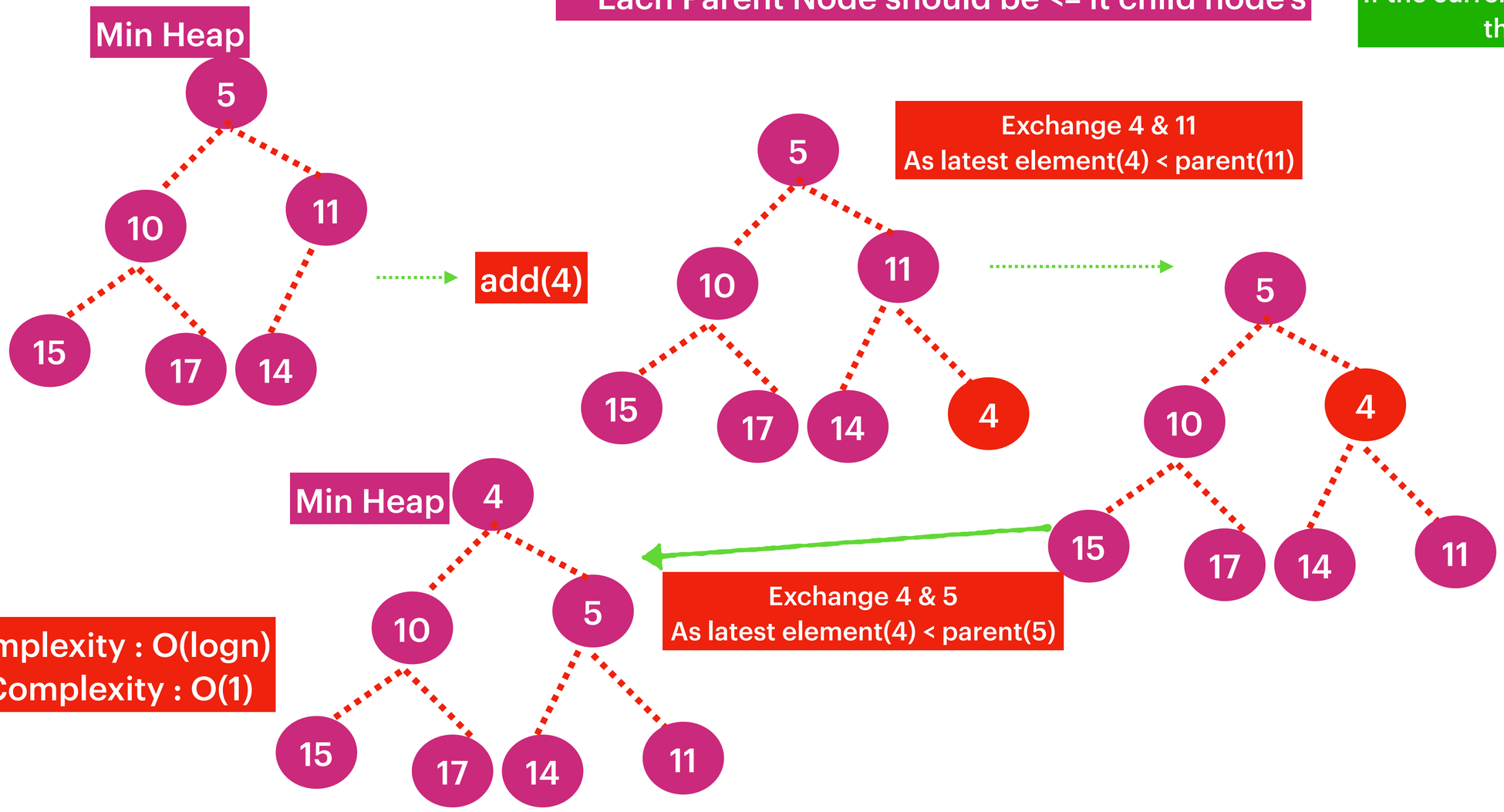
Min Heap

** The Data Structure should Complete Binary Tree

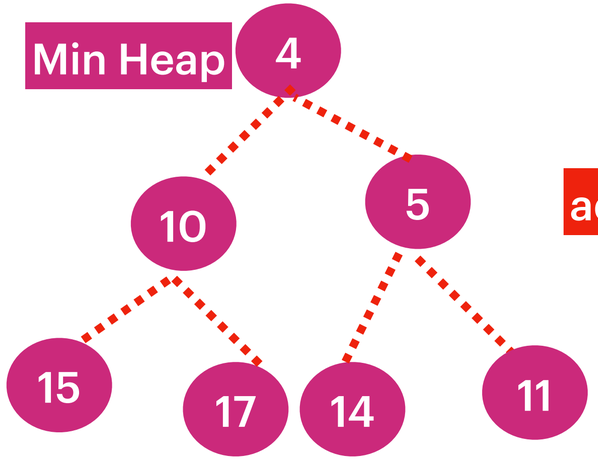
** Each Parent Node should be <= it child node's

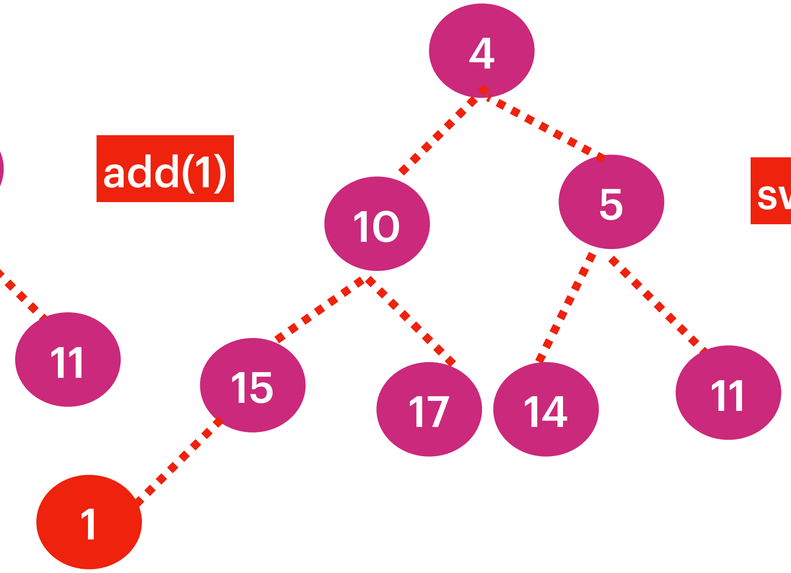
Algorithm :

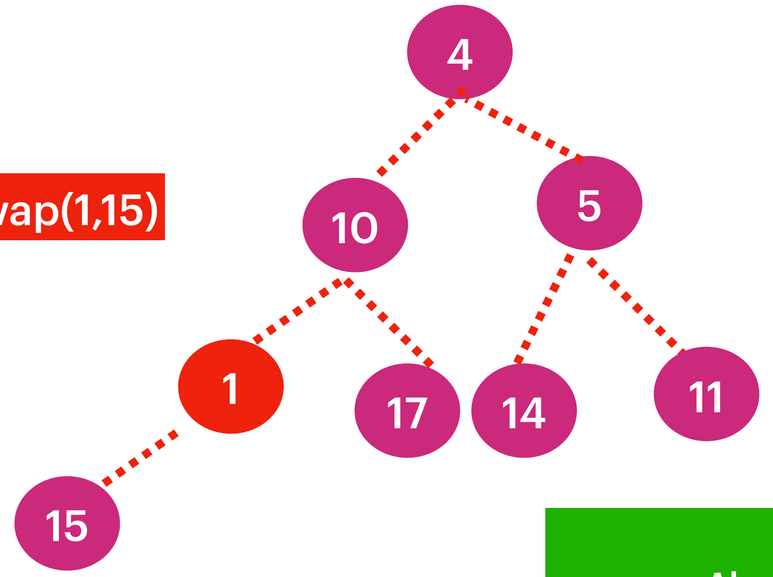
Add to the RightMost position.
If the current element > parent then swap.




Time Complexity : O(logn)
Space Complexity : O(1)

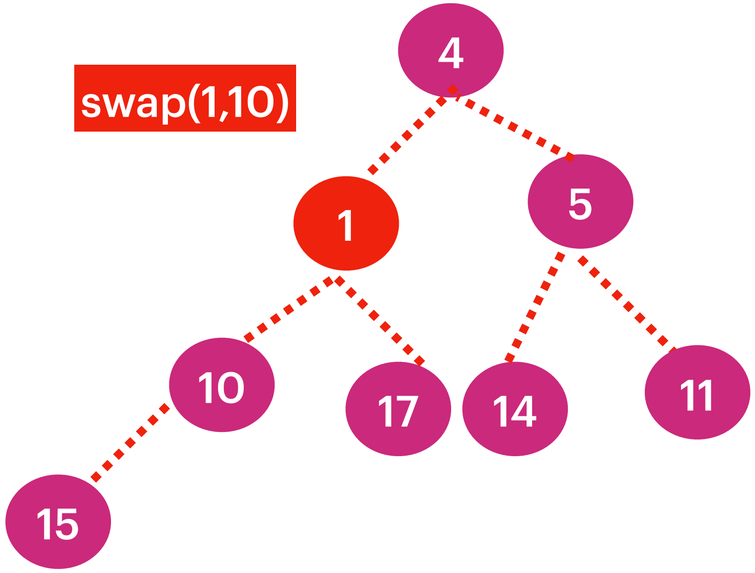


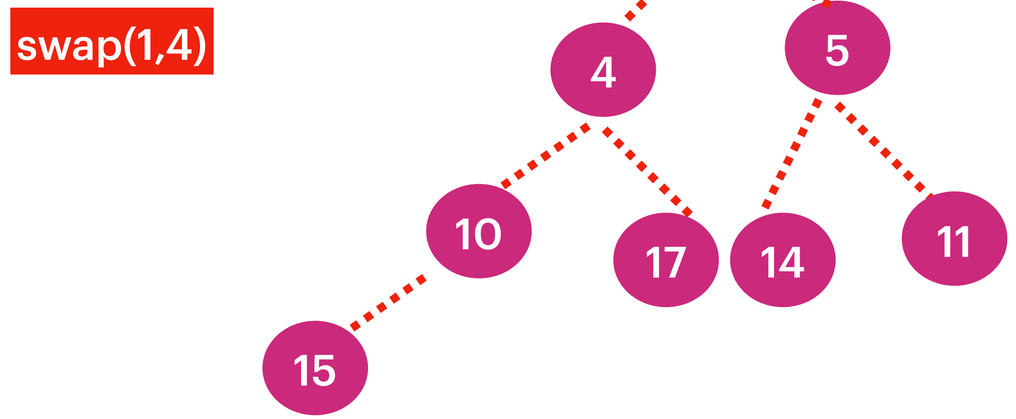




Algorithm :

Add to the RightMost position.
If the current element > parent then swap.

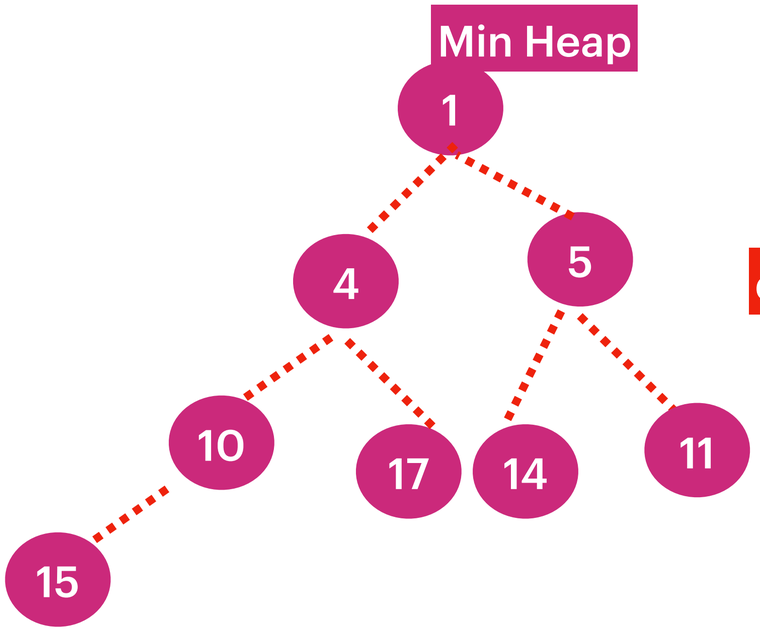


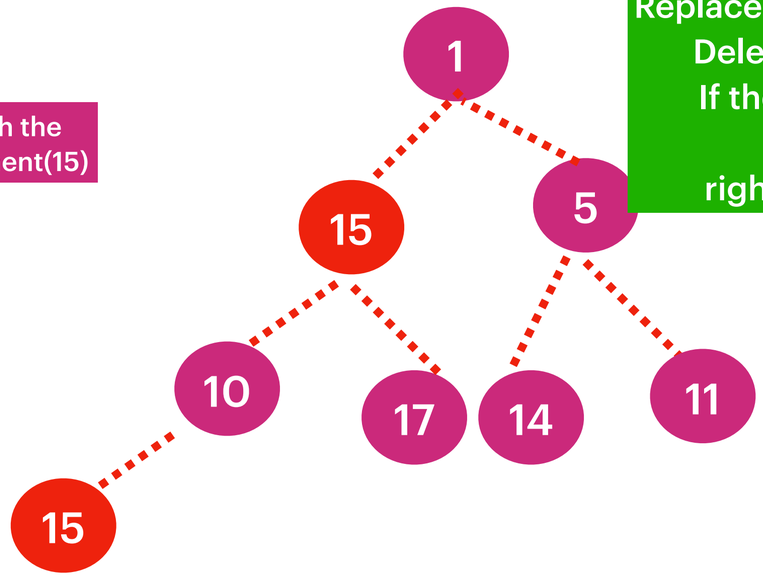
# Delete Operation On MinHeap

Algorithm :

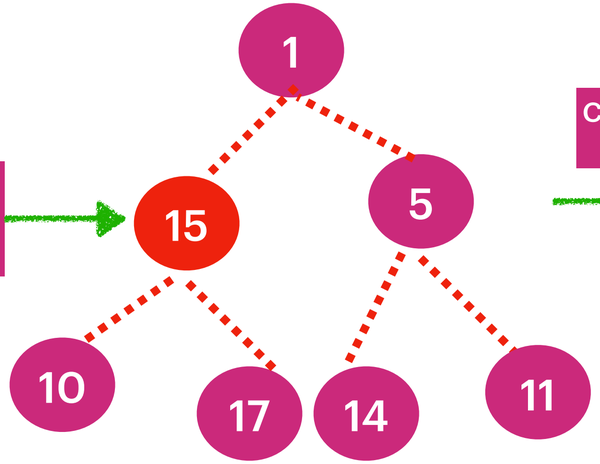
Replace with the right most element ,
Delete the right most element.
If the current Element either > (leftNode || rightNode) swap accordingly



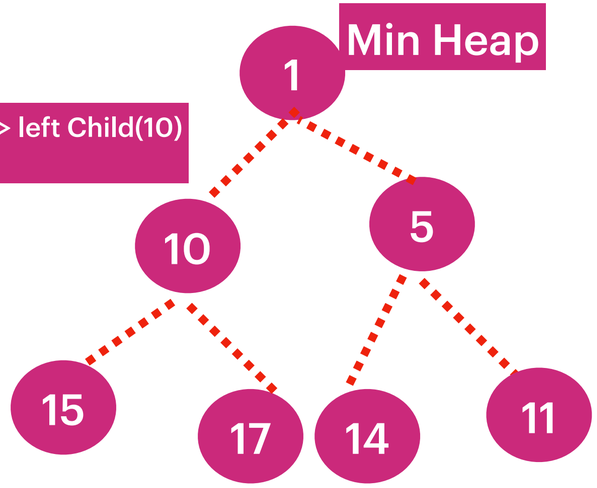

delete(4) → replace(4) with the Right Most Element(15)

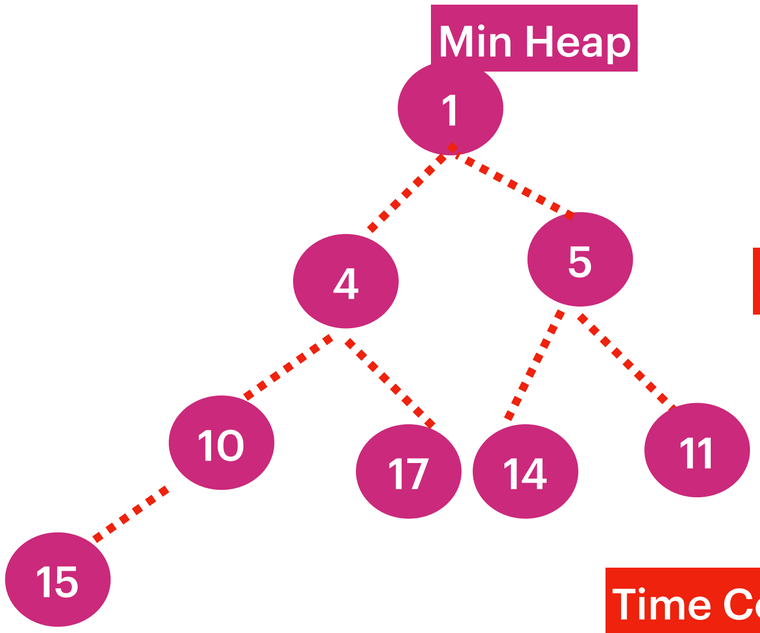



Delete the Right Most Element(15) in the max level of MinHeap



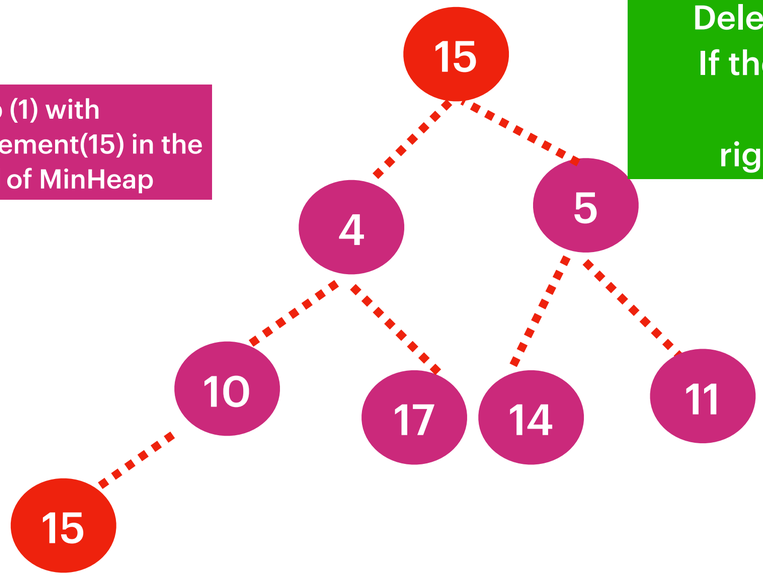

Current Parent(15) which is > left Child(10)
Swap (10,15)






## Delete Operation On MinHeap

delete(1) → Swap (1) with Right Most Element(15) in the max level of MinHeap

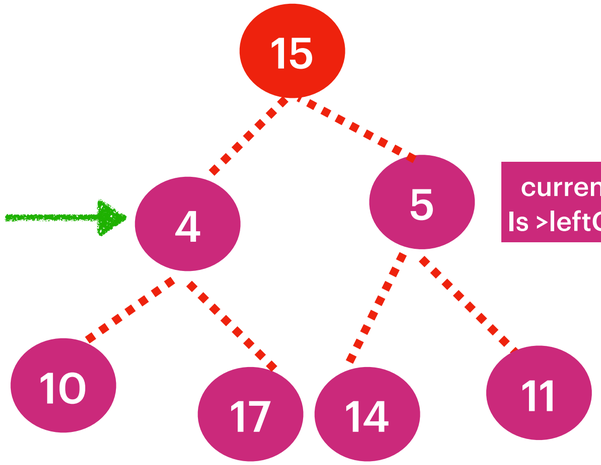



**Algorithm :**

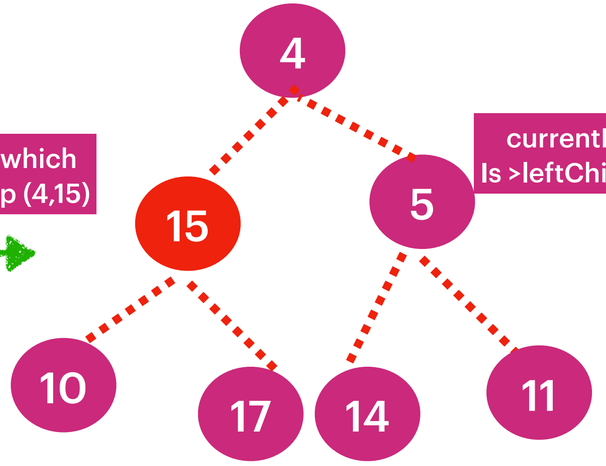
Replace with the right most element ,
Delete the right most element.
If the current Element either > (leftNode || rightNode) swap Iteratively

**Time Complexity : O(logn)**
**Space Complexity : O(1)**

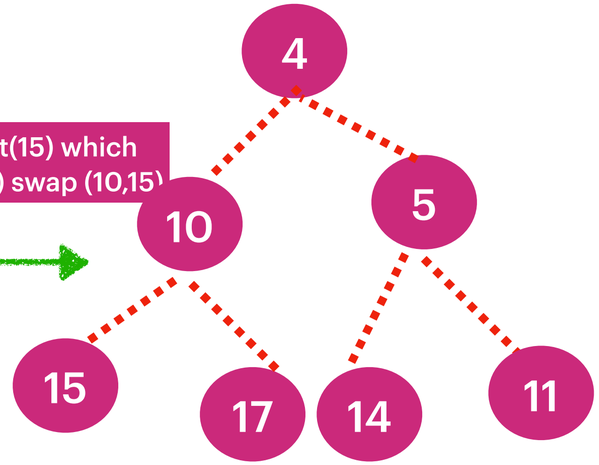
Delete the Right Most Element(1) in the max level of MinHeap →




currentParent(15) which Is >leftChild(4) swap (4,15) →



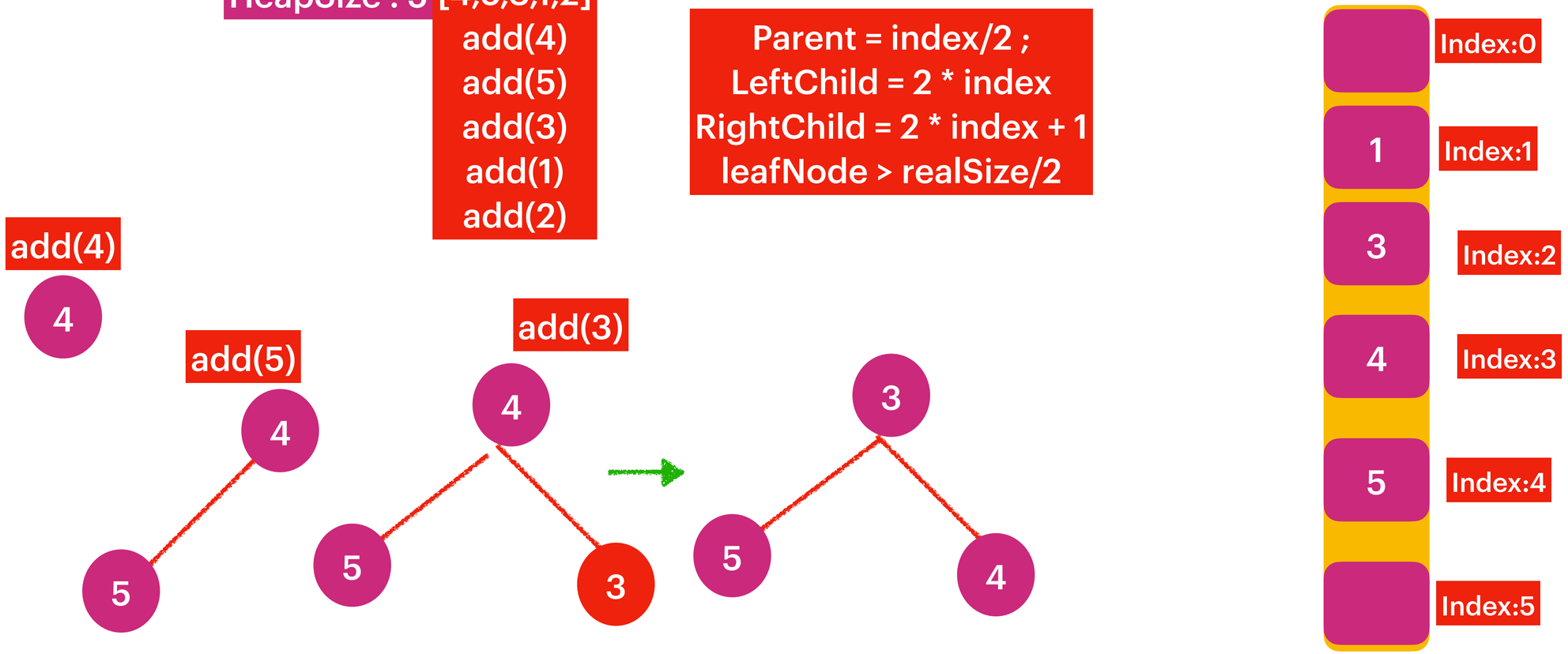

currentParent(15) which Is >leftChild(10) swap (10,15) →



HeapSize : 5 [4,5,3,1,2]

add(4)
add(5)
add(3)
add(1)
add(2)

Parent = index/2 ;
LeftChild = 2 * index
RightChild = 2 * index + 1
leafNode > realSize/2




add(4)

4

add(5)

4
5

add(3)

4
5
3

3
5
4

add(1)

3
5
4
1

3
1
4
5

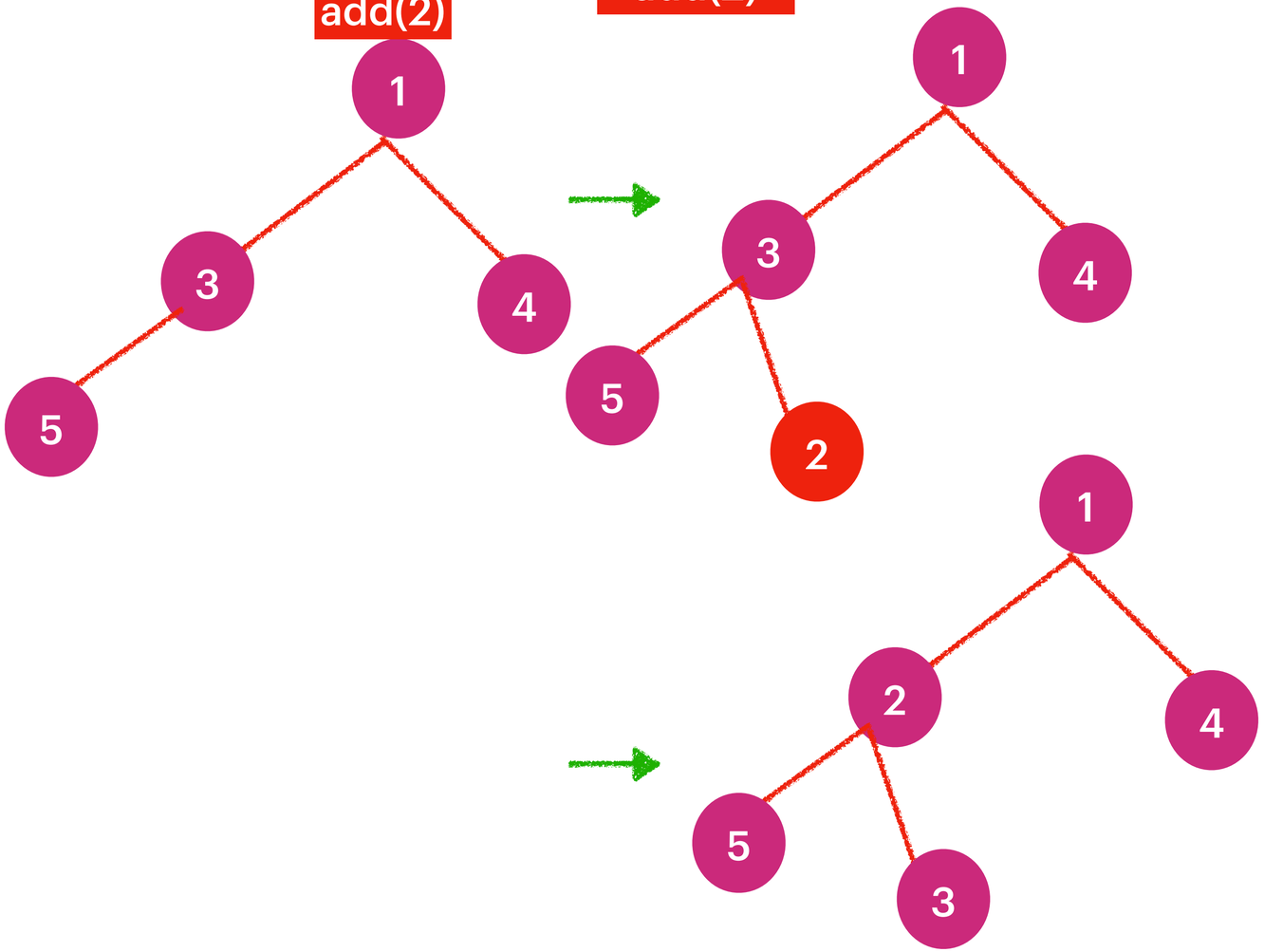
MinHeap

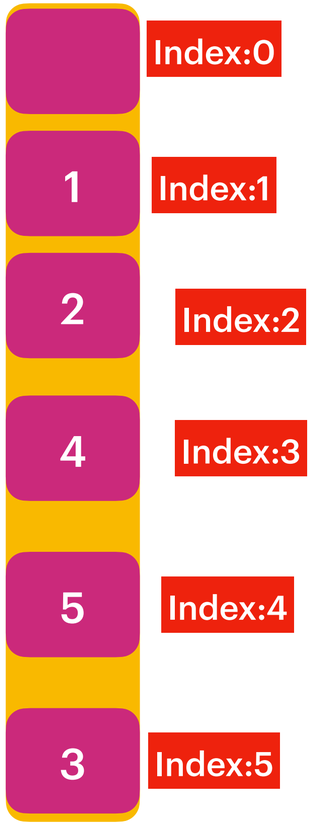
1
3
4
5

HeapSize : 5 [4,5,3,1,2]

add(4)
add(5)
add(3)
add(1)
add(2)

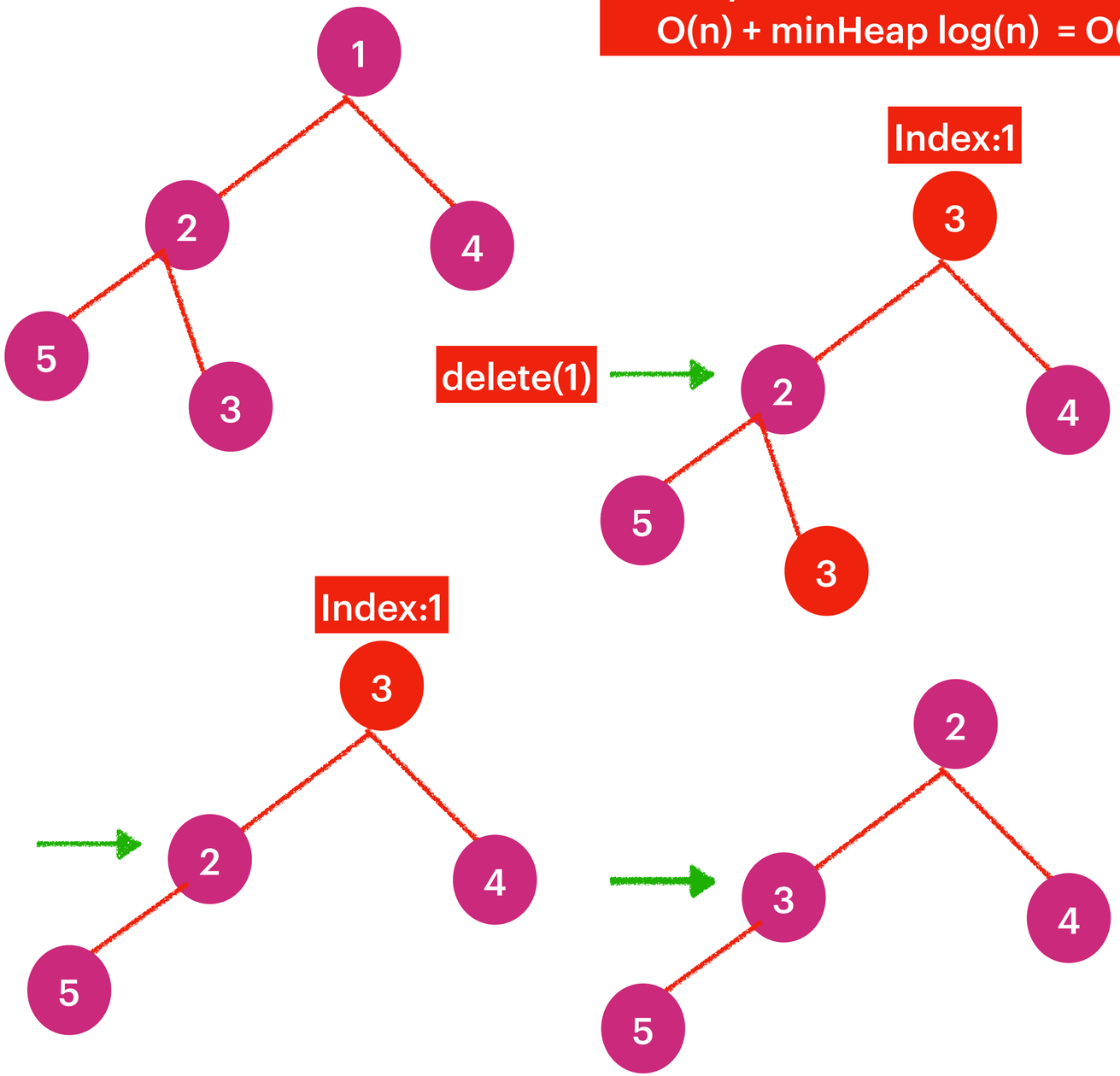
add(2)

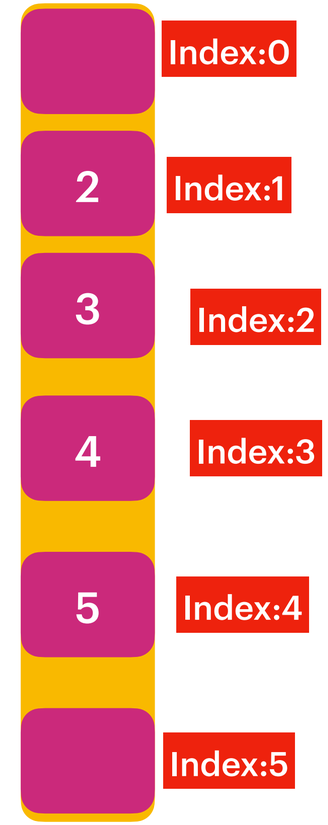



MinHeap

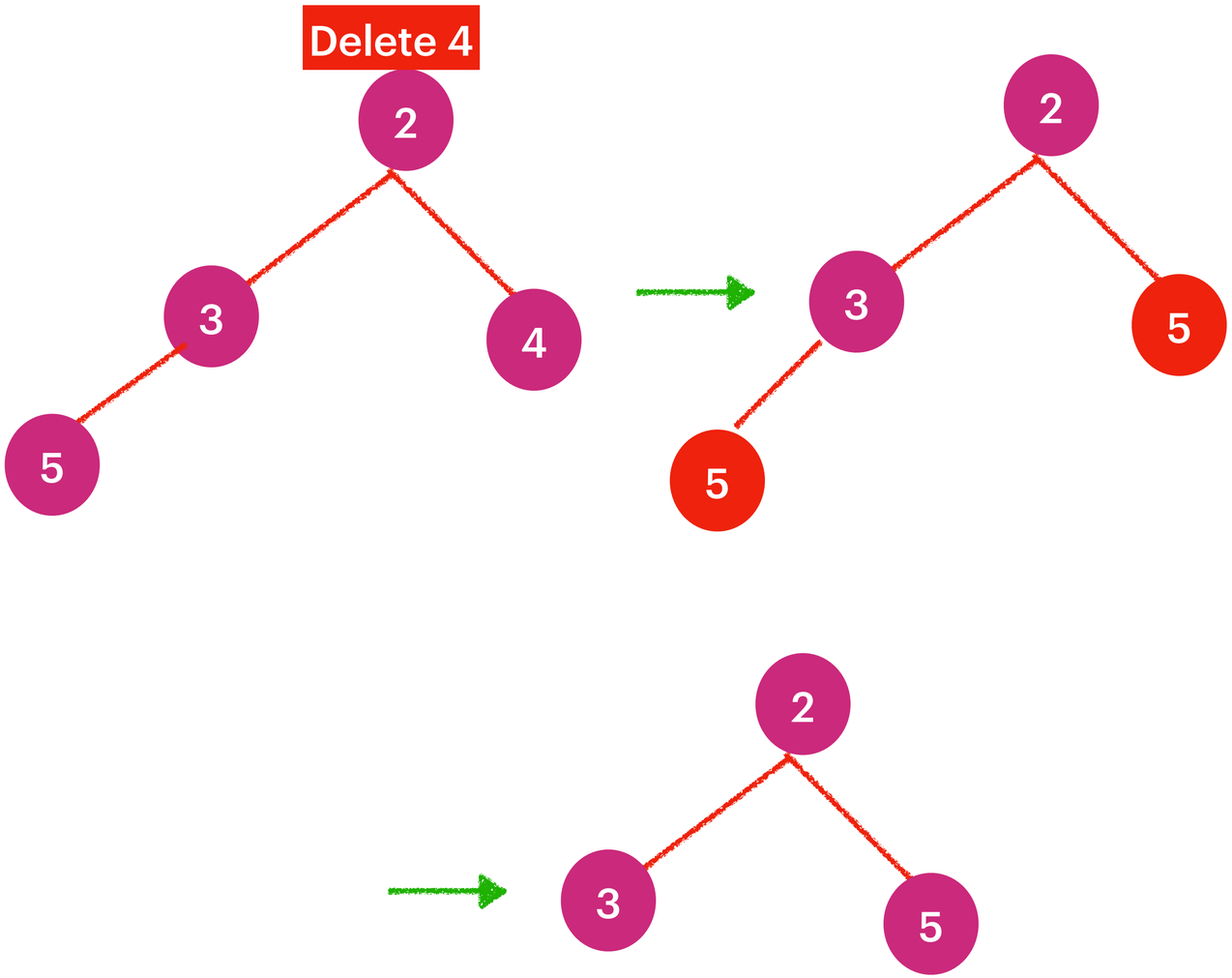
Identify the delete Element Index :
O(n) + minHeap log(n) = O(n)

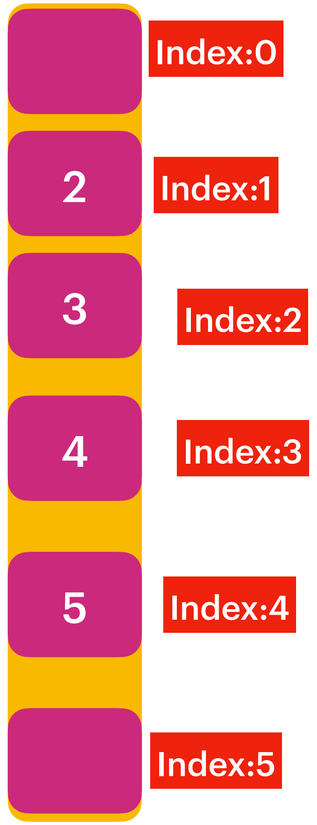



MinHeap

Identify the delete Element Index :
O(n) + minHeap log(n) = O(n)




MinHeap