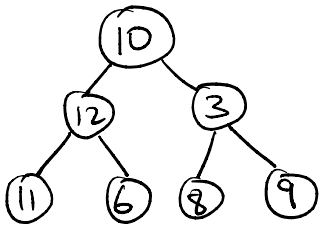
HW 3 - Part 4 Answers

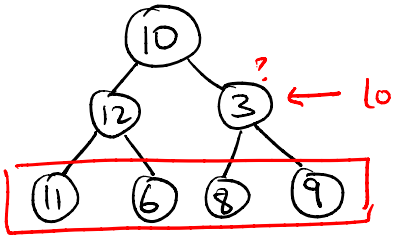
Dwaipayan Chonda

4) a. [10, 12, 3, 11, 6, 8, 9] can be shown in a binary tree as:



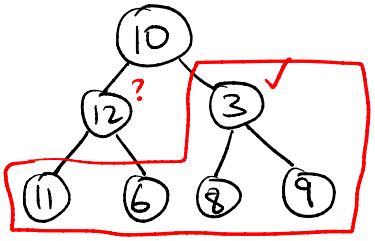
Now, buildHeap (using makeHeap algorithm) to make a minHeap

First, ignore all leaf nodes (which are heaps by definition).



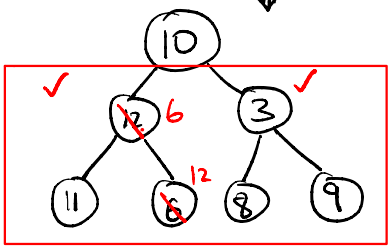
← look at last non-leaf node, call heapify()

ignore

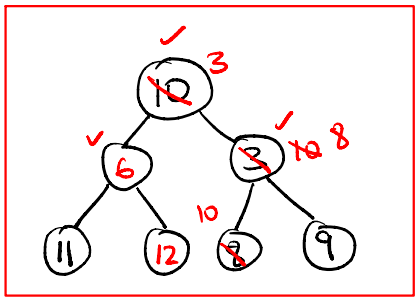


Satisfied

keep calling heapify on last non-leaf node.

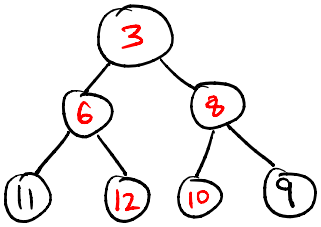


Satisfied



Satisfied

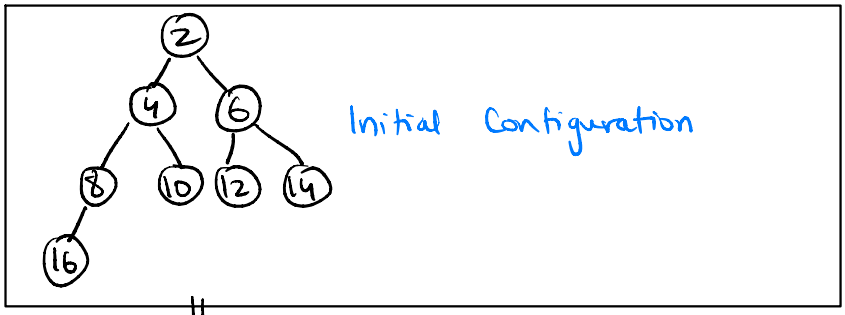
So, the final min-Heap is:



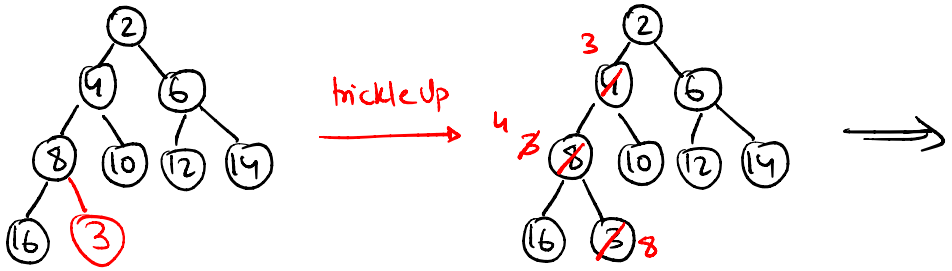
or [3, 6, 8, 11, 12, 10, 9]

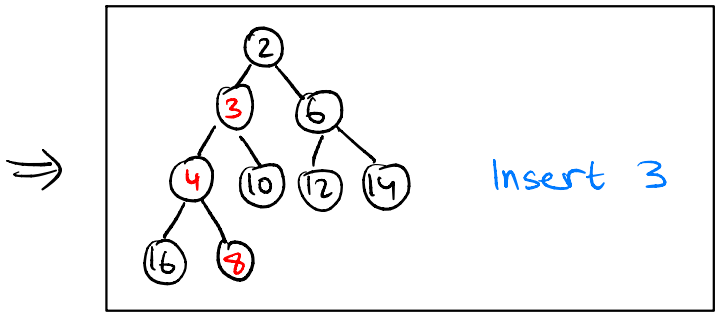
4) b. [2, 4, 6, 8, 10, 12, 14, 16]

Step (1)

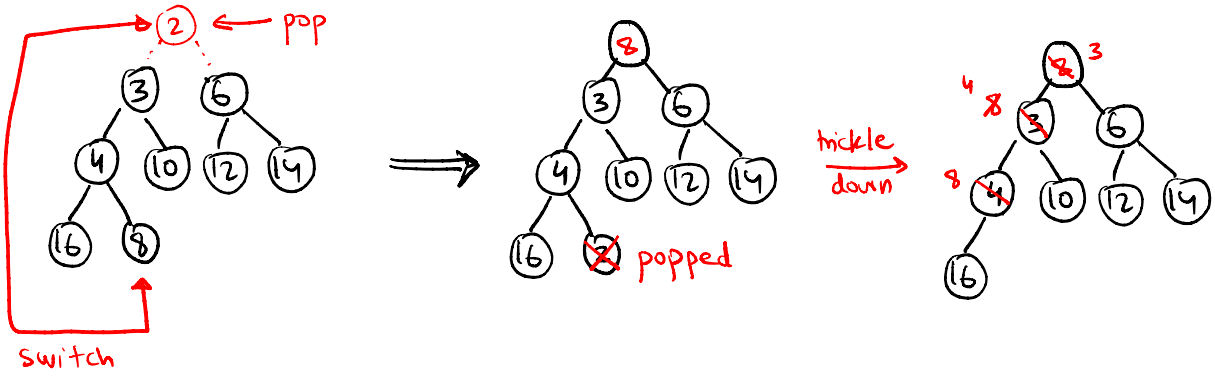


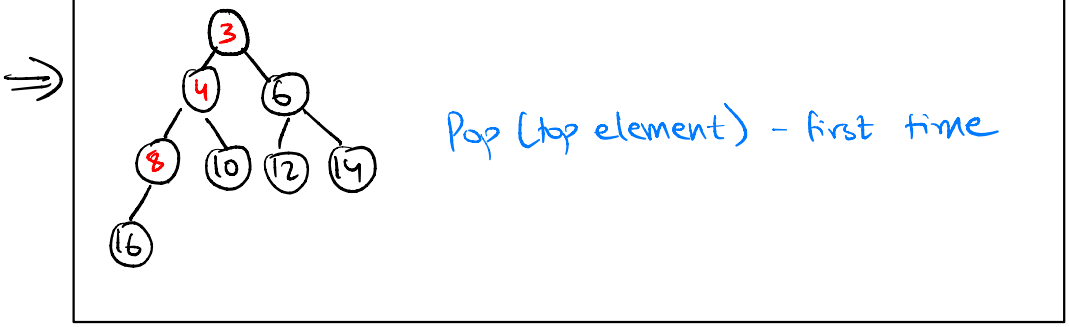
Step (2)



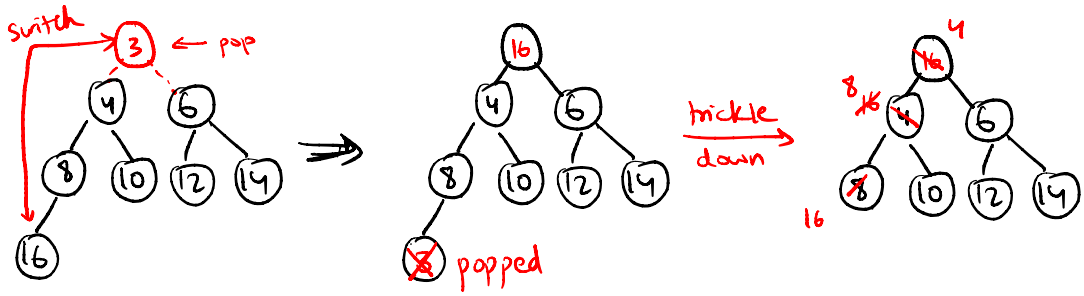


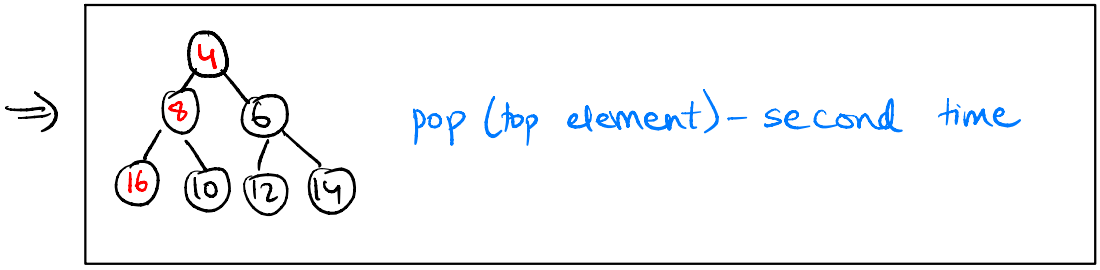
Step (3)



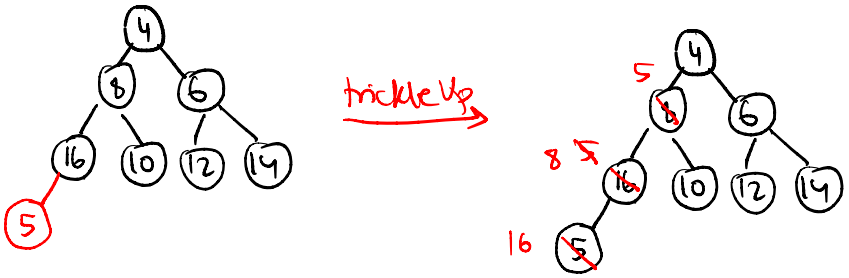


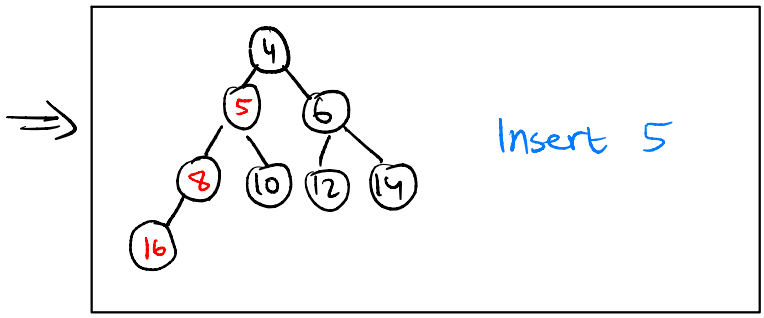
step 4


switch
3
pop
4
6
8
10
12
14
16
popped
trickle
down


4
8
6
16
10
12
14
pop (top element) - second time

step 5


4
8
6
16
10
12
14
5
trickle Up


Insert 5