

# Question: 1. An operating system execute process depending on their priority. Where highest (CO3,C3, number...

C [chegg.com/homework-help/questions-and-answers/1-operating-system-execute-process-depending-priority-highest-co3-c3-number-highest-priori-q57052453](https://www.chegg.com/homework-help/questions-and-answers/1-operating-system-execute-process-depending-priority-highest-co3-c3-number-highest-priori-q57052453)



1. An operating system execute process depending on their priority. Where highest number has highest priority. Consider at a particular instance of time, 10 processes arrives with following priority. Now build a Binary Heap using the priority number. [CO3,C3, Mark: 6]

Process	Priority	Process	Priority
P1	3	P6	3
P2	5	P7	4
P3	2	P8	5
P4	4	P9	1
P5	6	P10	6

- b. Now delete the top four priority patients from the heap and perform necessary operations to rebuild the heap after deletion. [CO3,C3, Mark: 4]

show the step by step process of these binary heap ( code is not needed)

Show transcribed image text

## Expert Answer (i)



Anonymous answered this  
Was this answer helpful?

110 answers

Priority queue binary heap

Ways of implementing a priority queue

Priority queues are usually implemented with heaps since this gives them the best possible time complexity.

Priority queue is an abstract data type hence heap are not the only way to implement priority queue. As an example, we could use an unsorted list but this would not give us the best possible time complexity.

- A binary heap is a binary tree that supports the heap invariant. In a binary tree every node has exactly two children.
- A complete binary tree is a tree in which at every level, except possibly the last, is completely filled and all the nodes are as far left as possible.

Binary heap has two properties

- Min heap: the parent node should be always less than child node.
- Max heap: the parent node should be always greater than child node.

Rules to insert an element to the binary heap tree.

- Always insert the element to the last position of the tree.
- After insertion heapify the tree.

a)  $P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10}$   
 3 5 2 4 6 3 4 5 1 6

1) First insert 3.

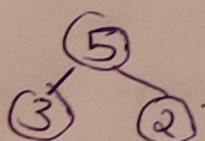


insert 5.

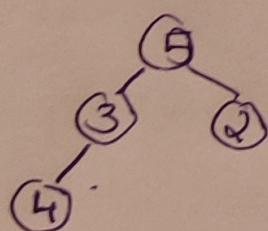
Since 5 is greater than 3 swap.



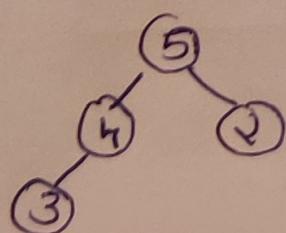
2) Insert 2.



3) Insert 4.

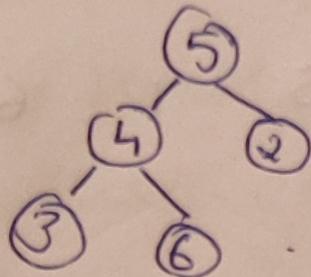


Since  $4 > 3$  swap.

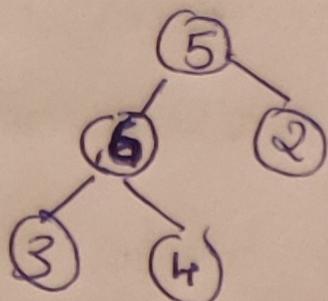




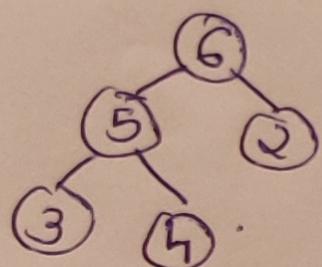
5. Insert 6.



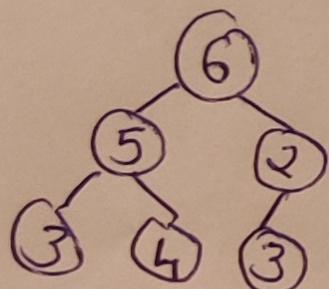
Since  $6 > 4$  swap .



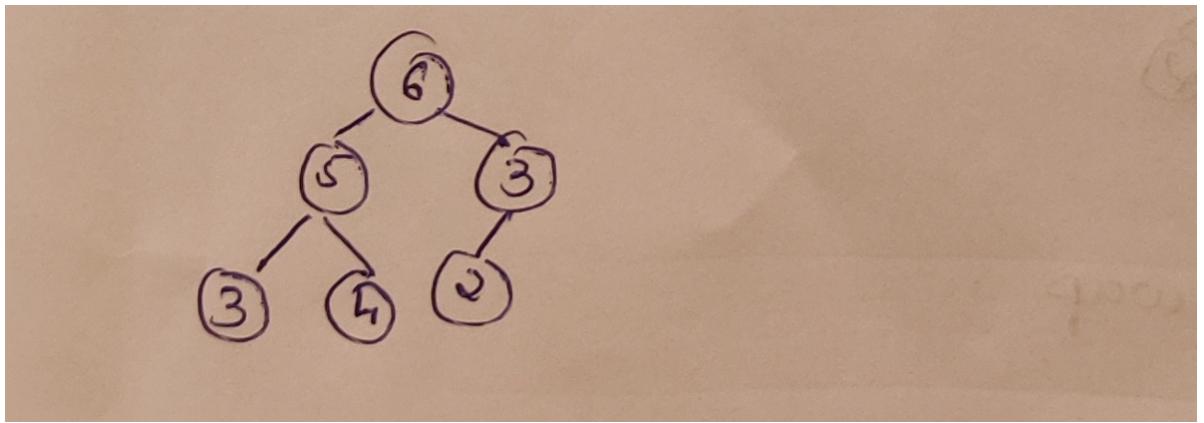
again  $6 > 5$  swap .



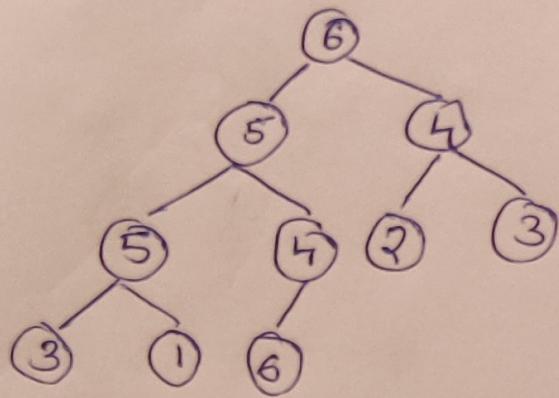
6. Insert 3.



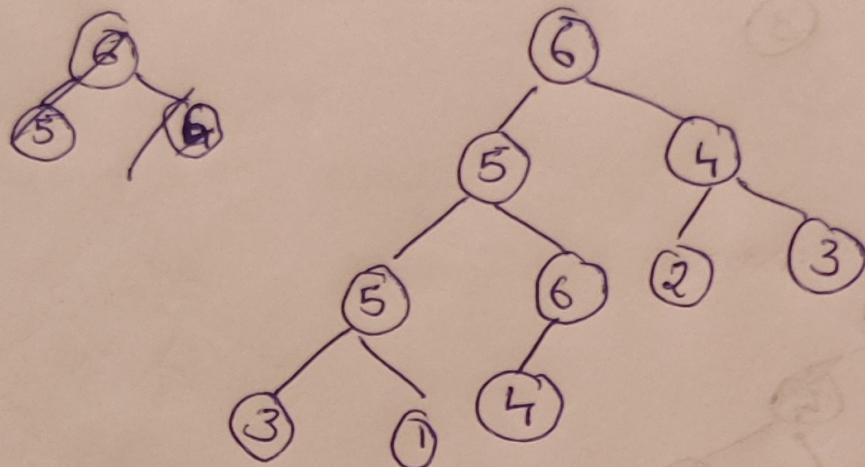
Since  $3 > 2$  swap .



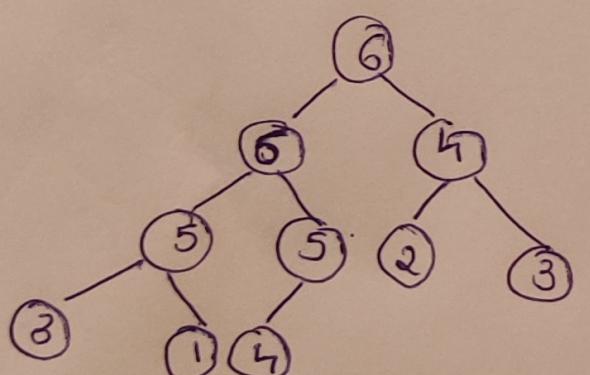
10. insert 6.



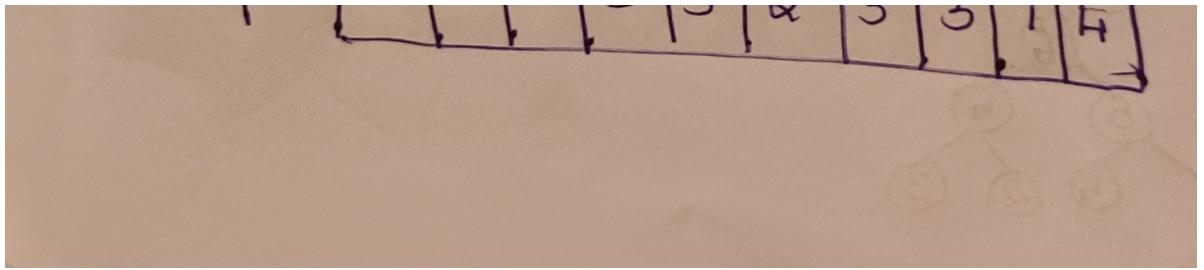
Since  $6 > 4$  swap.



Since  $6 > 5$  swap.



heap. [ 6 | 6 | 4 | 5 | 5 | 2 | 2 | 1 | 1 ]



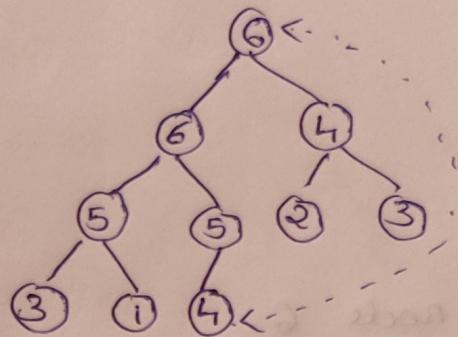
Rules to delete node from priority queue binary heap.

1. Select the node to be deleted.
2. Swap that node with the last leaf node.
3. Delete the last leaf node. (i.e the node to be deleted.)
4. Heapify the heap tree and priority queue.

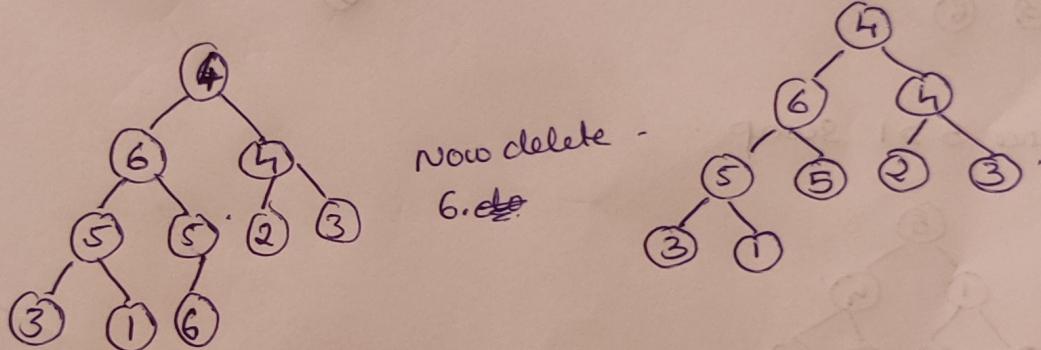
8.

Now we have to delete 6, 6, 4, 5

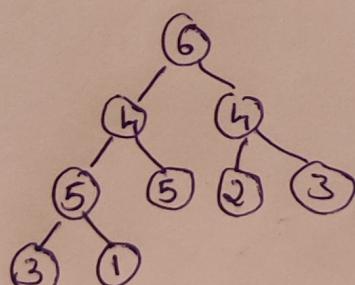
we have



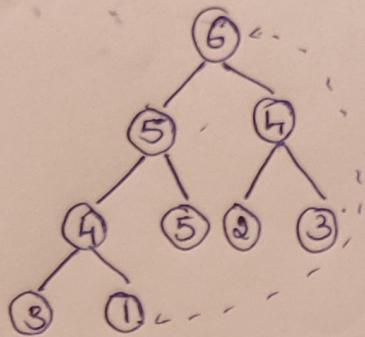
10. To delete 6 we have to swap the last node ie (4) and insert 4 in place of 6 and delete the node .



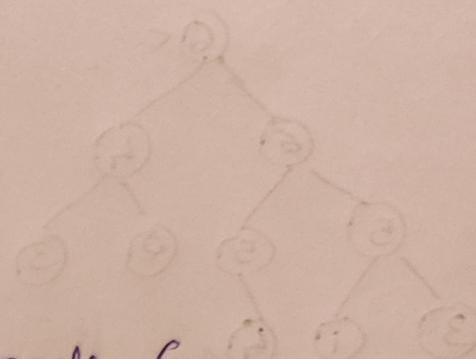
Now swap 6 and 4 since  $6 > 4$ .



Since  $5 > 4$  swap

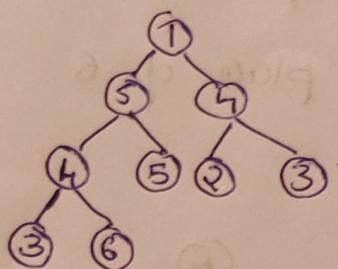


steps of search in sorted array

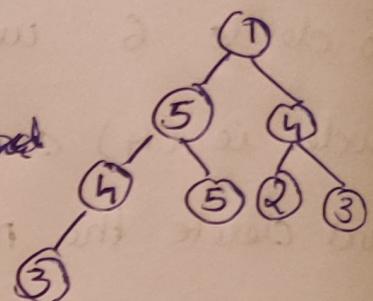


2. Now delete next node 6.

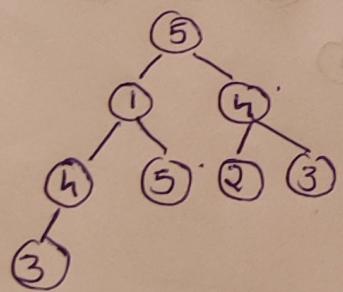
swap 6 and 1.



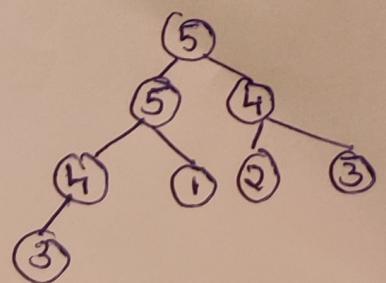
now delete 6



Since  $5 > 1$  swap

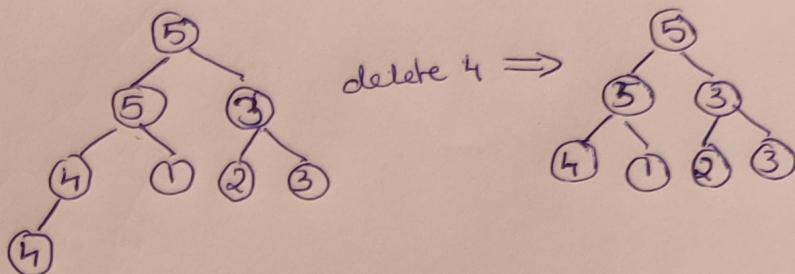


Since  $5 > 1$  again swap



3. Now delete node 4.

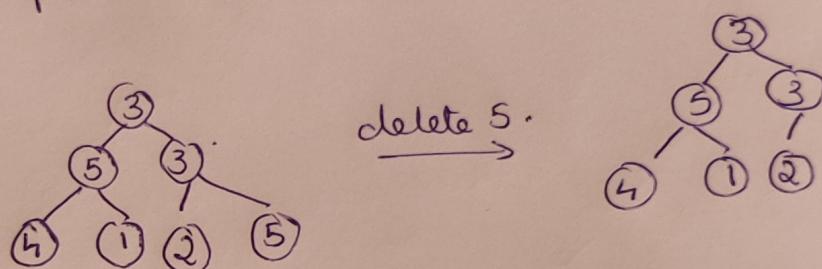
To delete node we have to swap 4 and 3.



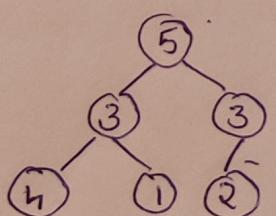
~~Ans~~

4. Delete 5.

Swap 3 and 5.



swap 3 and 5 since  $5 > 3$ .



swap 3 and 4 since  $4 > 3$ .



heap. 5 4 3 3 1 2

Comment

