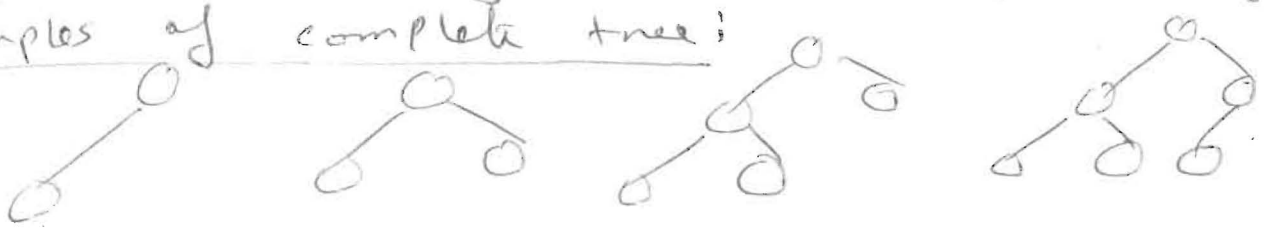
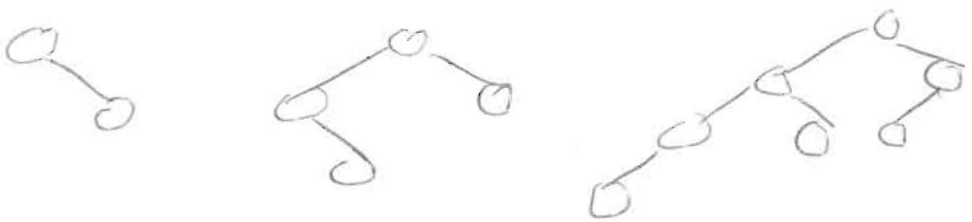
Heap

- Heap is an array represented by binary tree.
- The binary tree should be completed tree.
- When you fill up a tree as complete tree you fill it level by level from left to right.

Examples of complete tree:



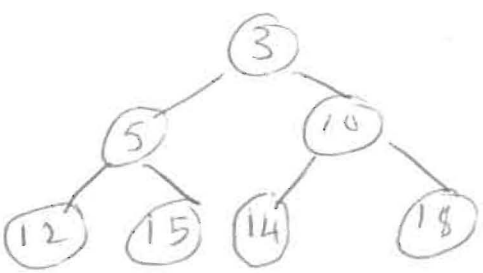
following not complete tree:



if you want to have minimum heap, you should have every node less than it's child(ern).

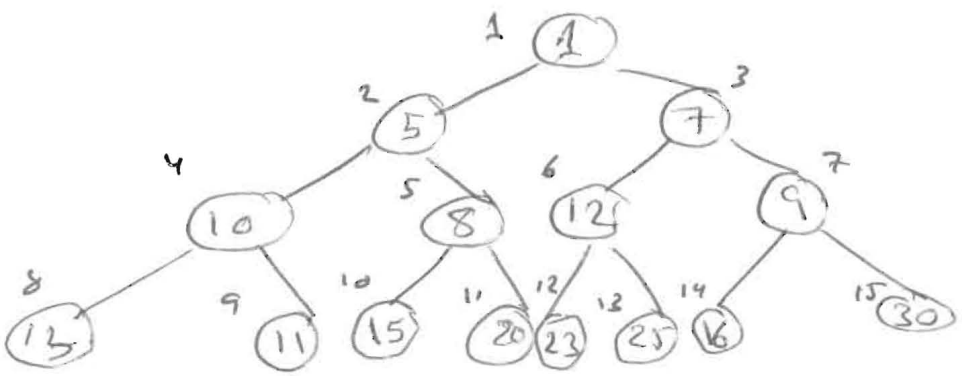
if you want to have maximum heap you should have every node more than it's child(ern).

Example of minimum heap:-



in minimum heap always the smallest key is the root, in maximum heap the largest value always in root.

how to put the values of complete binary tree in an array as heap.



	1	5	7	10	8	12	9	13	11	15	20	23	25	16	30
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

every node has location i and its

left child in $(2*i)$

and right child in $(2*i)+1$

for Example node $i = 4$ left child $2*i = 8$

right child $= (2*i)+1 = 9$ as you see in the above Binary tree.

if you know a node number and you want to know it's parent location in the Array:

the node is $i/2$ (as int).

for Example $i = 11$ it's parent $= 11/2 = 5$.

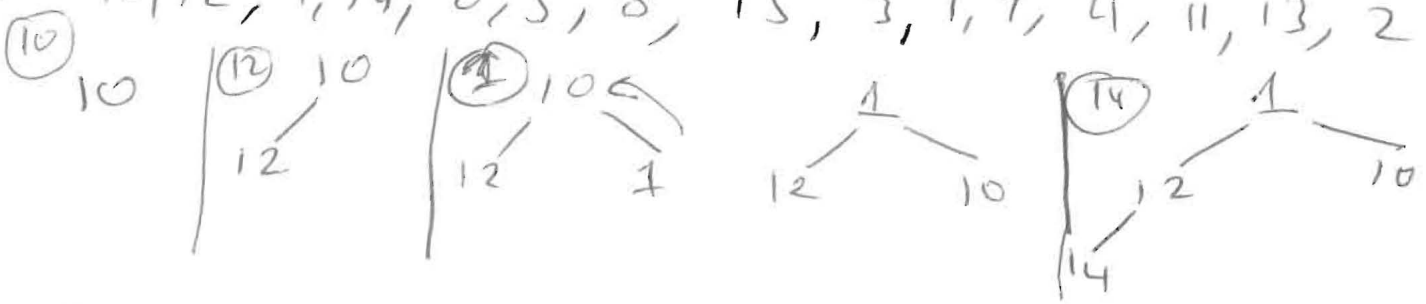
- always the last parent index in the heap is size of heap / 2. Like in the above tree size = 15 so last parent is 15/2 = 7.

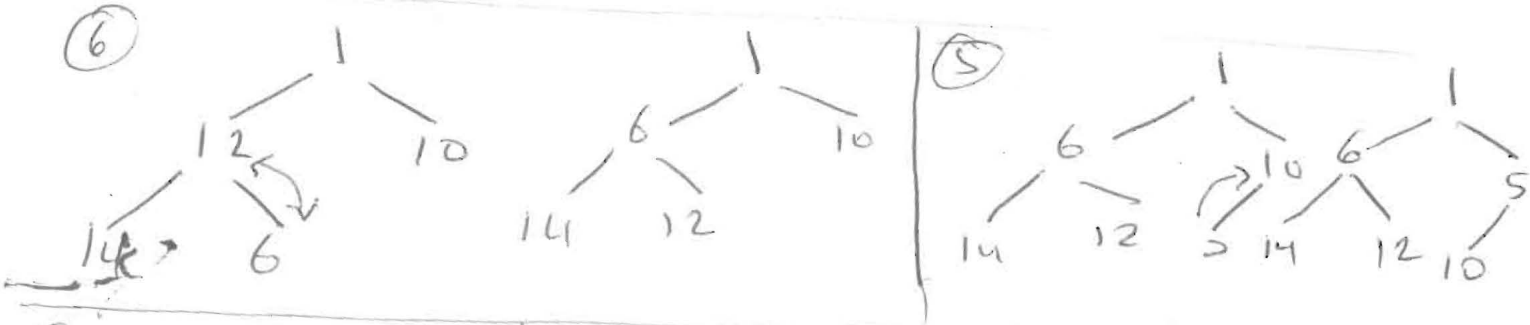
To insert in a heap using binary tree always add at the end as complete tree then do siftup for example

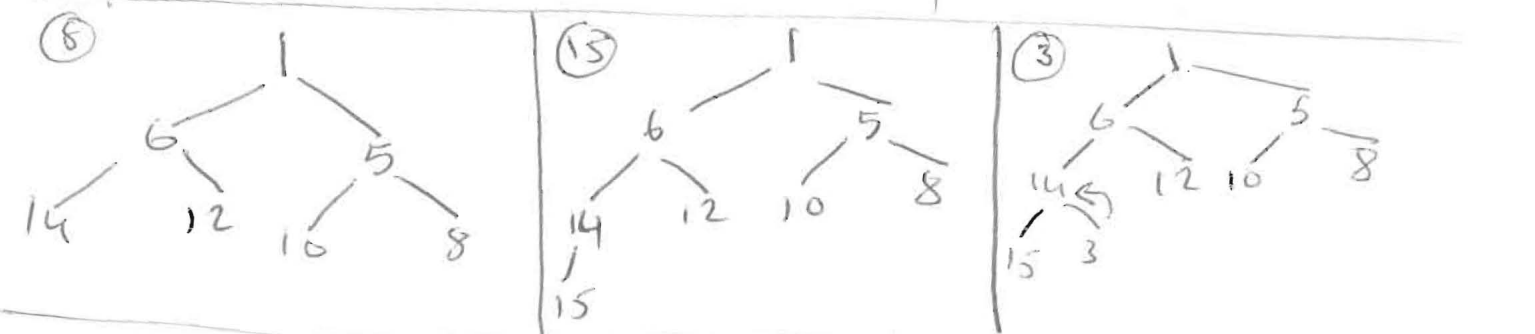
insert the following elements in a heap:

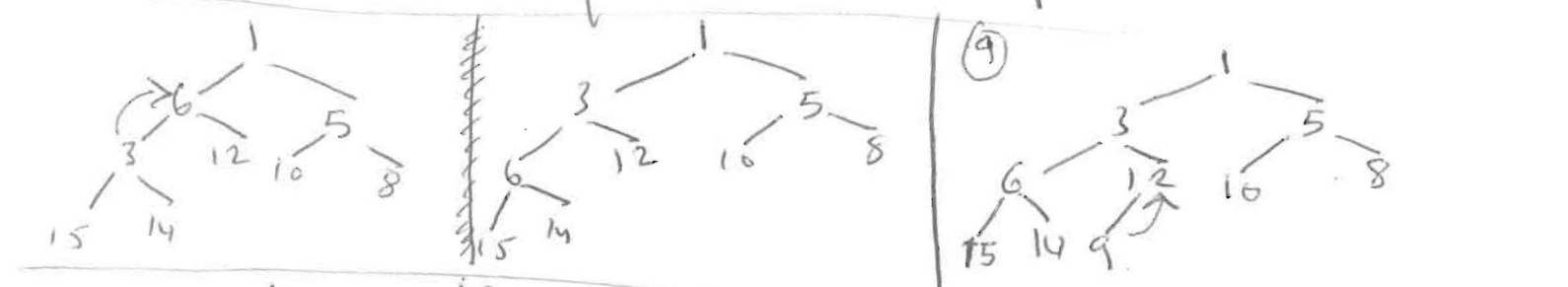
minimum heap

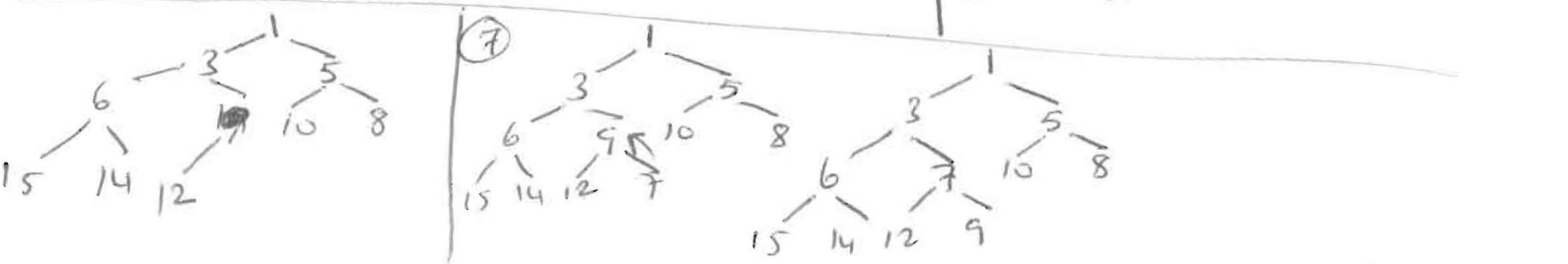
10, 12, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13, 2





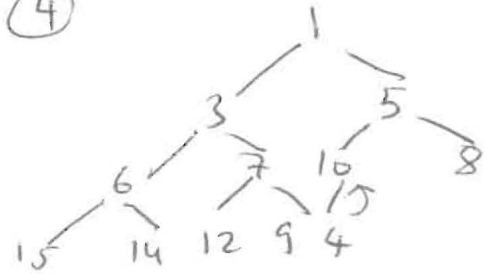


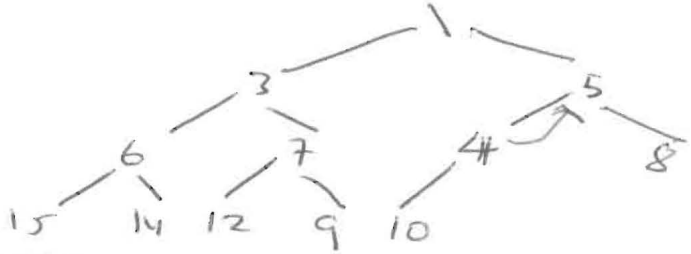


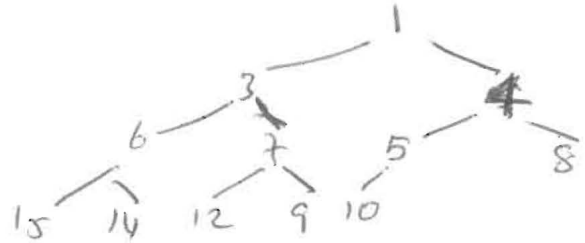


⇒

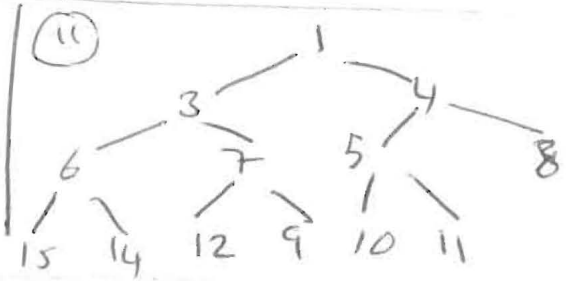
(4)



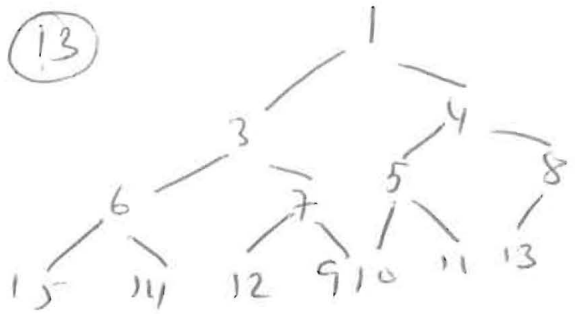




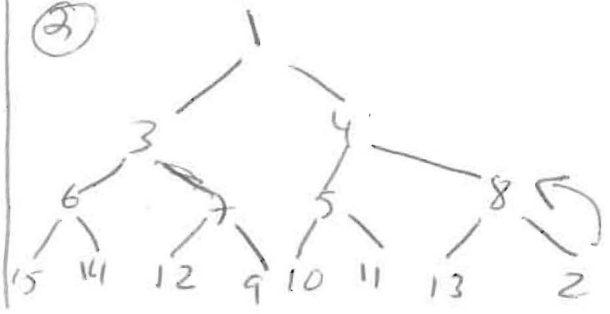
(11)

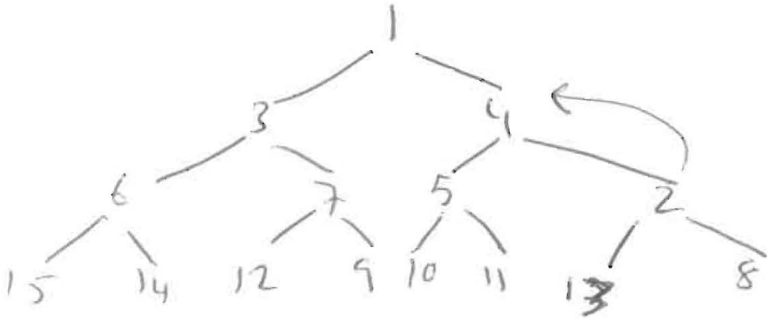


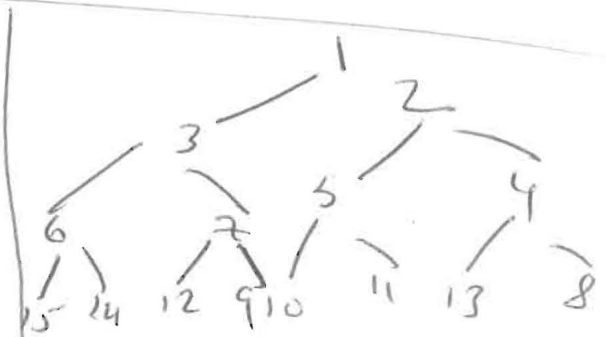
(13)



(2)





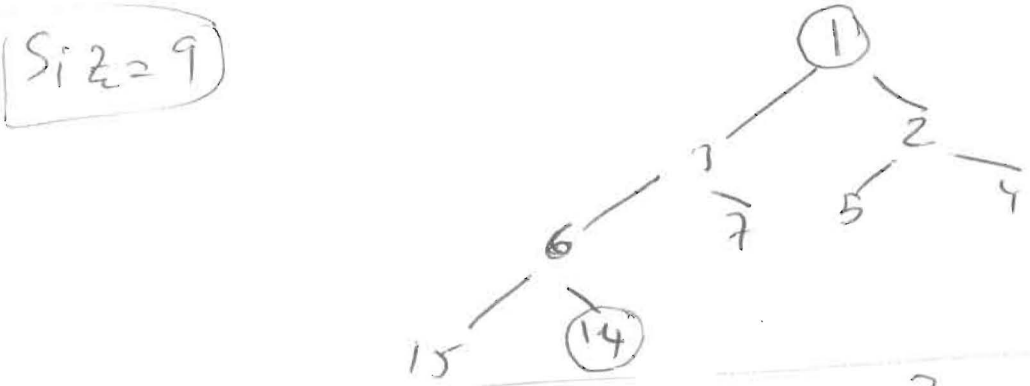


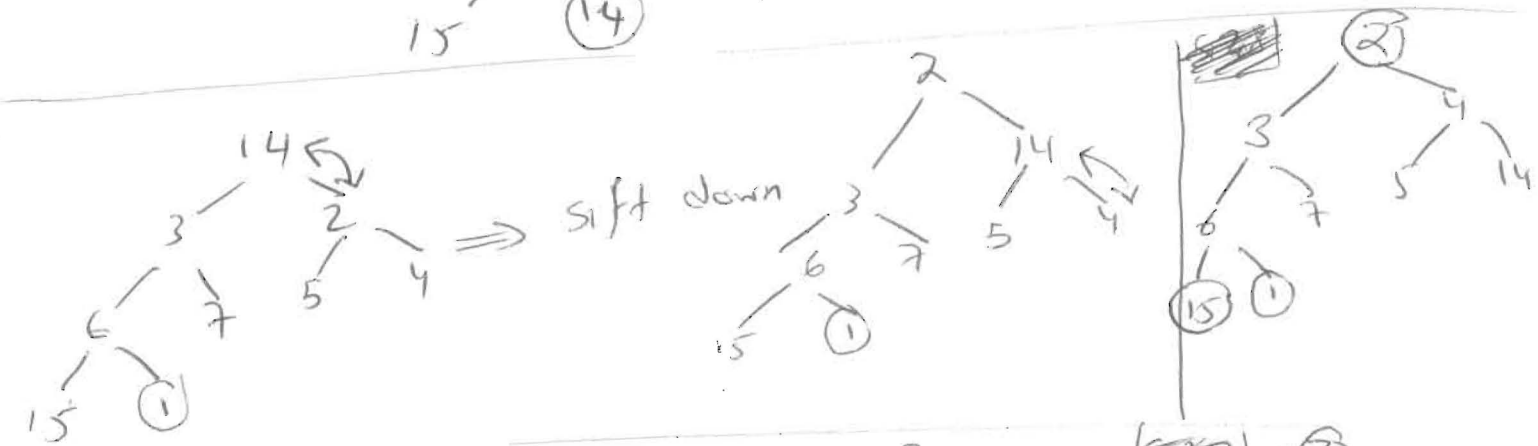
We can use the heap with one type of sort which called heap Sort.

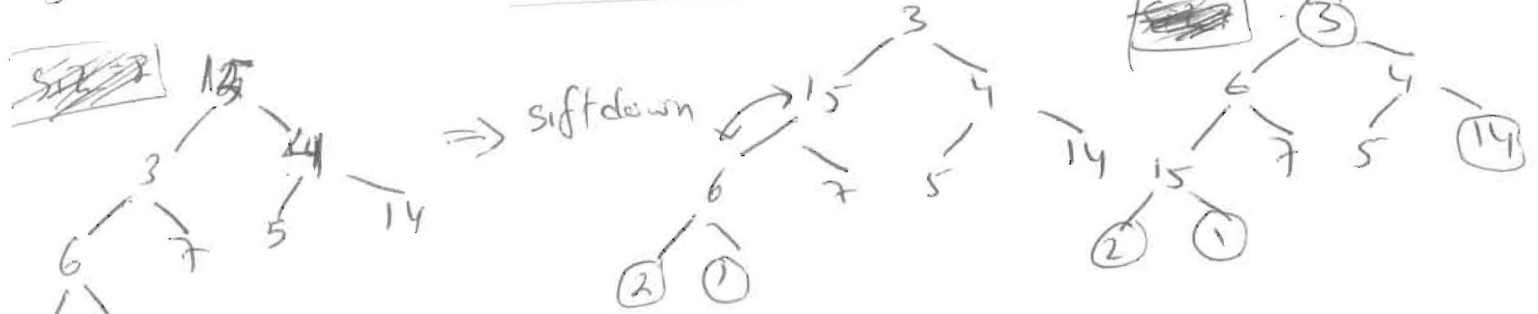
when you do heap sort you swap the root with the last element, decrease size by 1 then sefldown.

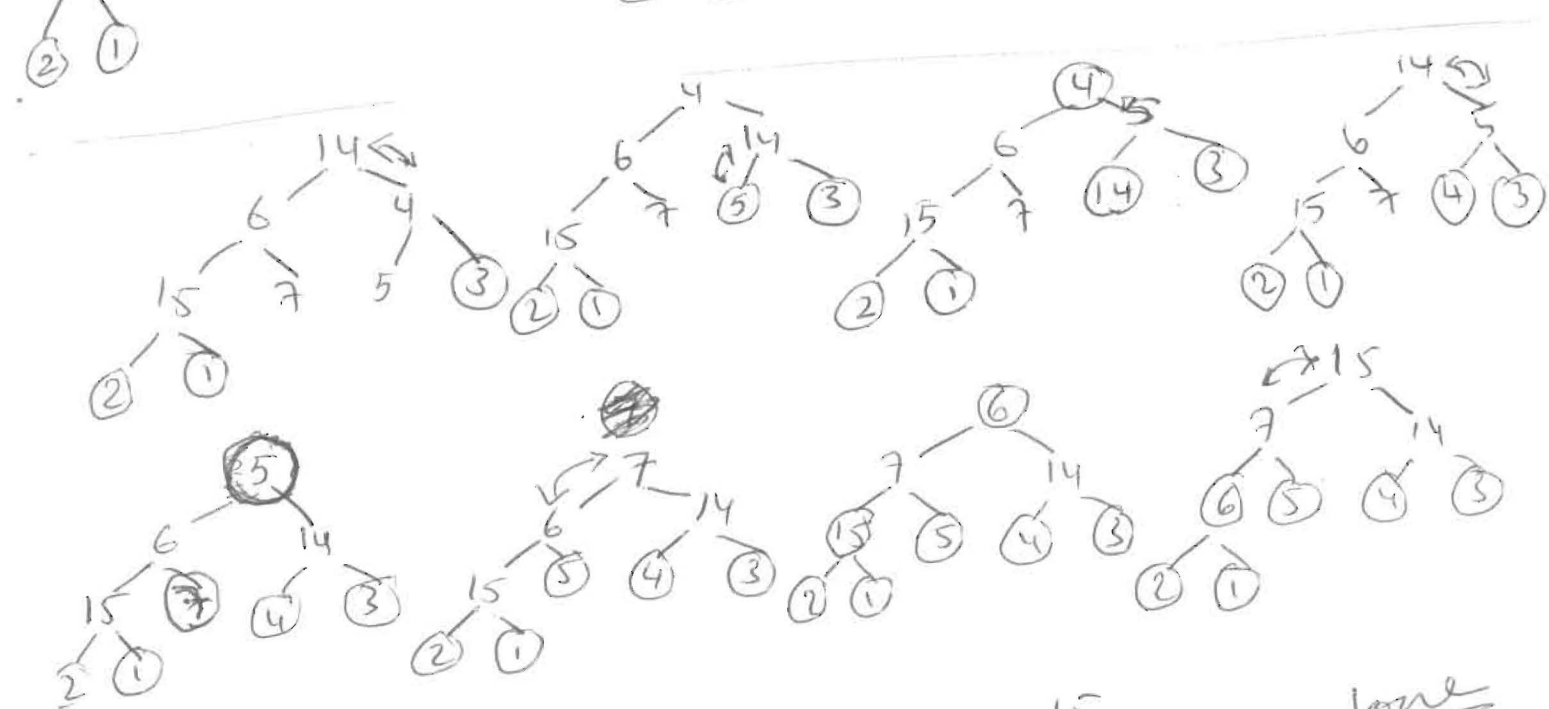
for Example:

Size = 9

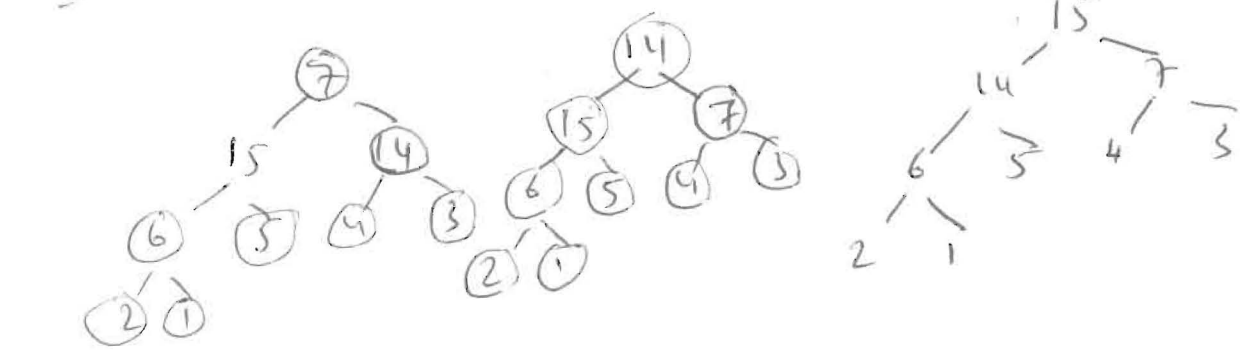








done



if you want to sort in increase order you must have maximum heap.
in decreasing order you mush have minimum heap.

You can use priority queue with heap implementation.

enqueue : (1) insert.
(2) sift up.

dequeue (serve) : (1) delete root.
(2) sift down.
