ASSIGNMENT #6

SUBMITTED BY :

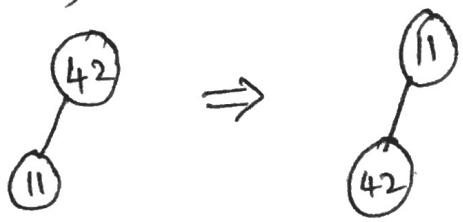
NAME : SAI KIRAN TADURI

NET ID : SXT161730

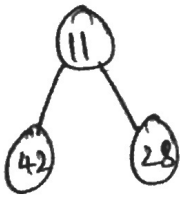
(1) 42, 11, 28, 8, 8, 61, 18

Insert 42,

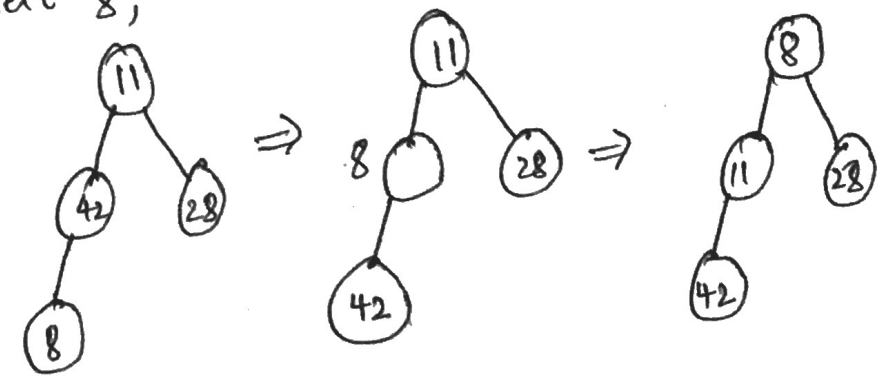
Insert 11,



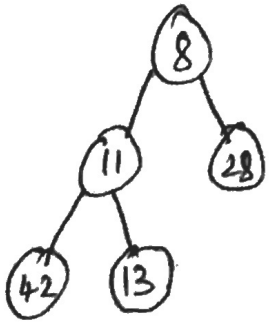
Insert 28,



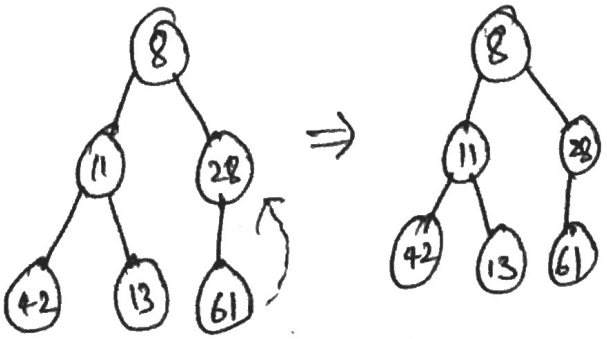
Insert 8,



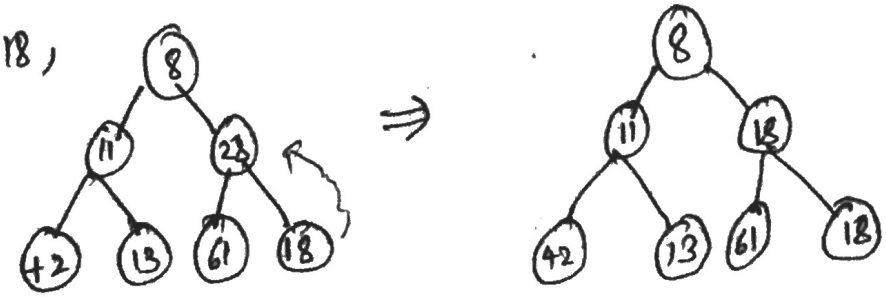
Insert 13,



Insert 61,



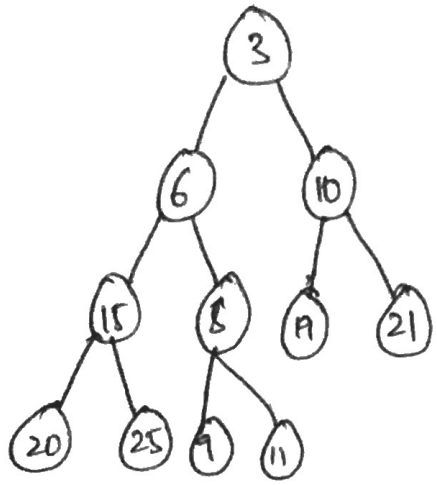
Insert 18,



(2)

	8	11	18	12	13	61	28		
0	1	2	3	4	5	6	7	8	9

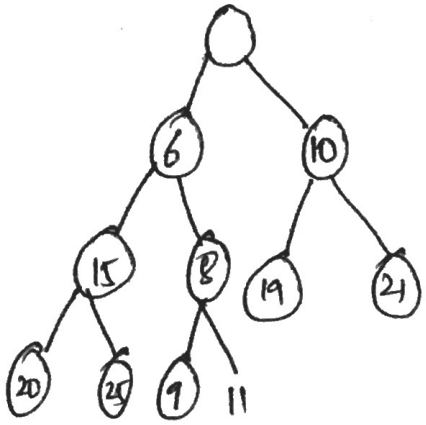
(3)



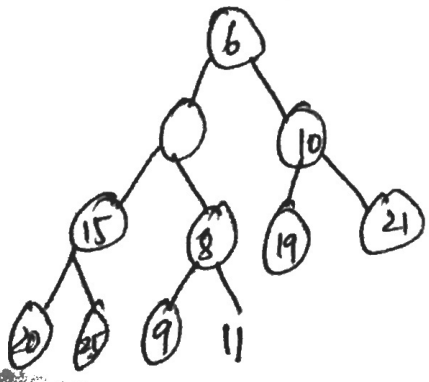
Performing DeleteMin on the above heap.

DeleteMin of 3,

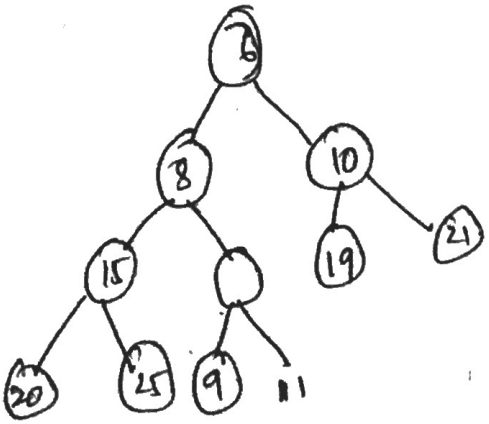
Creates hole at root, test 11.



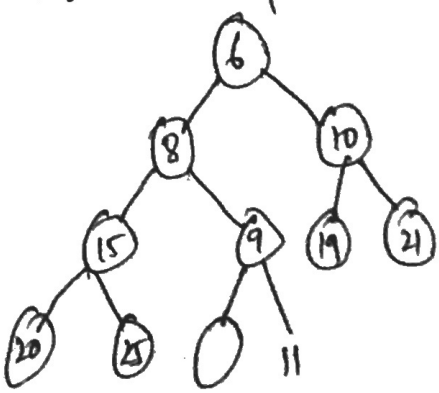
Slide 6 up,



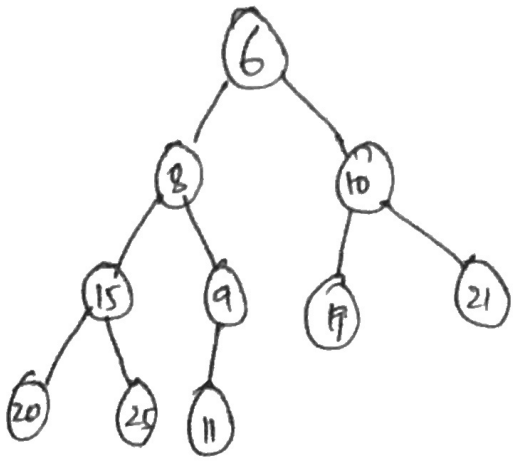
Slide 8 up,



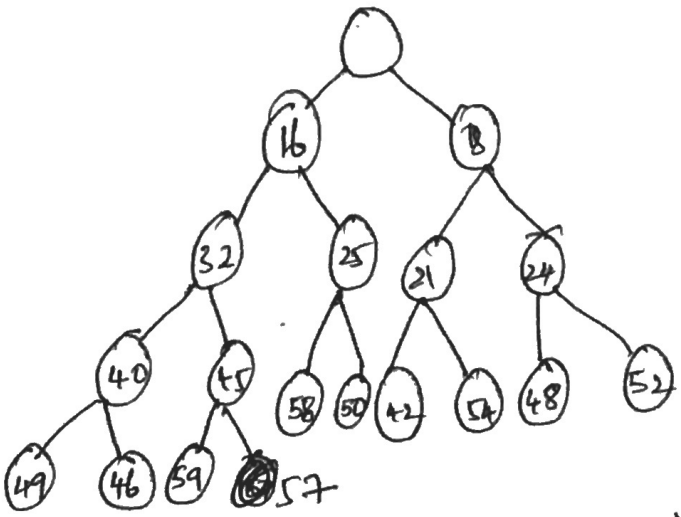
Slide 9 up,



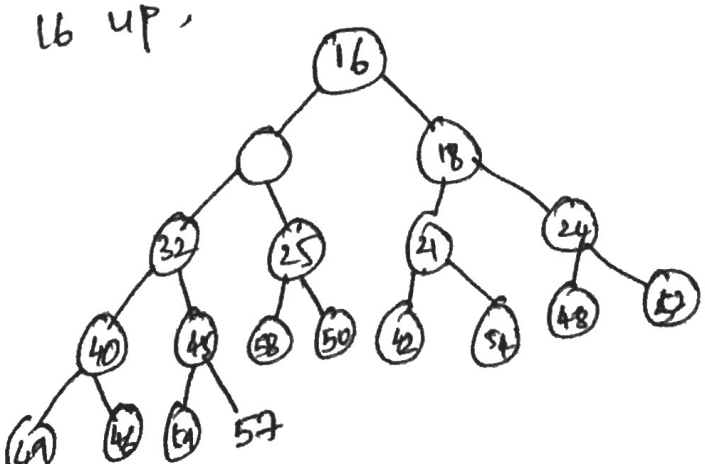
Put 11 in hole,



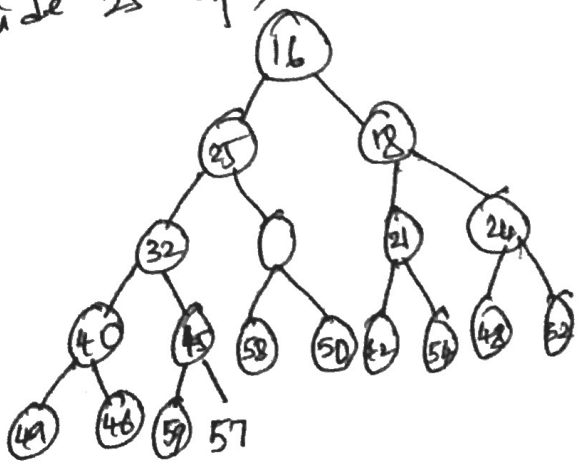
(4) Performing deleteMin on binary heap.
deleteMin of 12,
Creates a hole at root, test 57



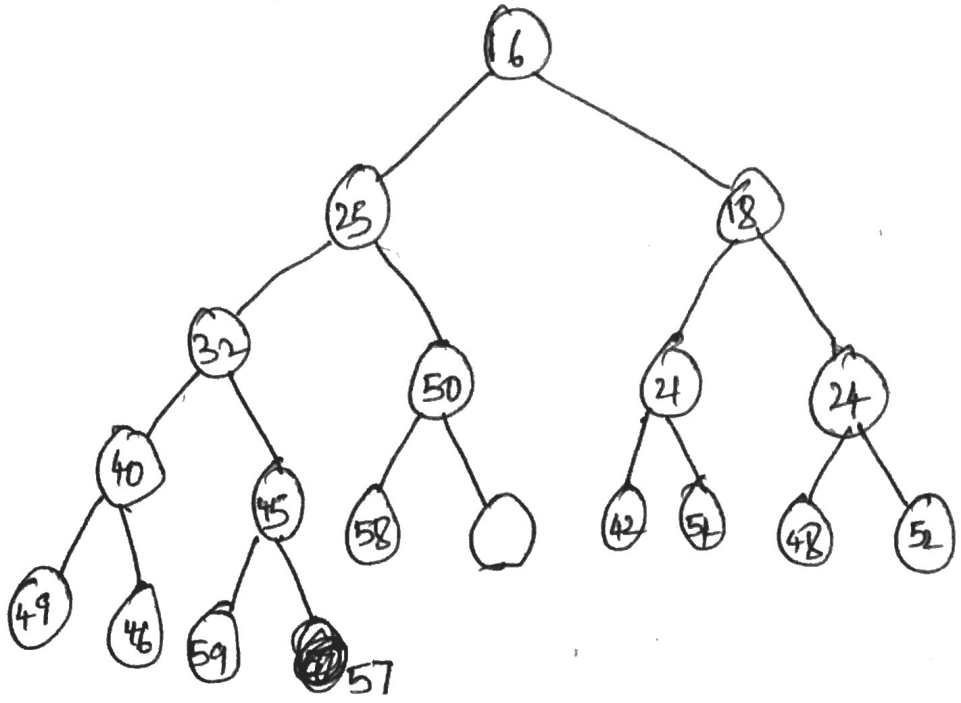
slide 16 up,



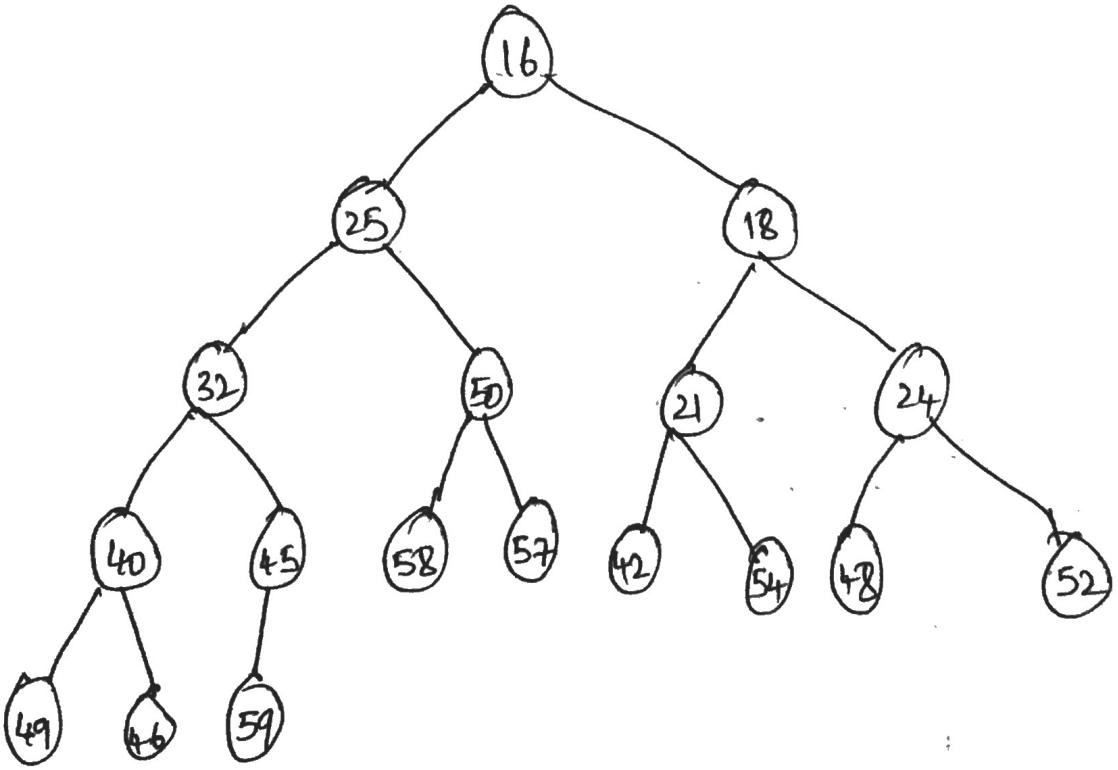
slide 25 up,



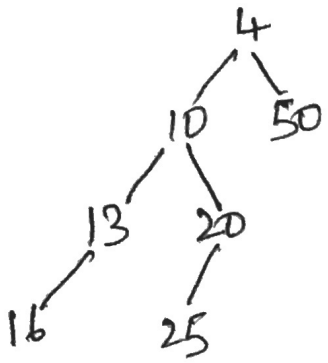
slide 50 up,



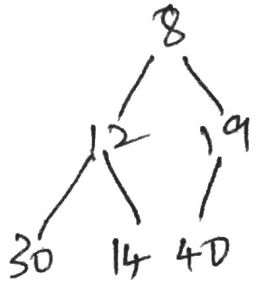
Put 57 in hole,



h1



h2

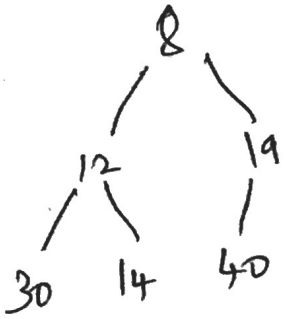


h1 has smaller root, so merge h2 with right subtree of h1

h1

50

h2

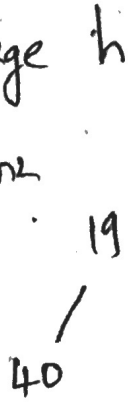


h2 has smaller root, so merge h1 with right subtree of h2

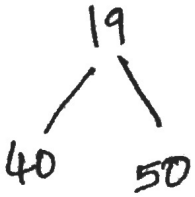
h1

50

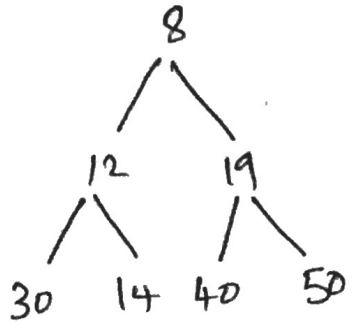
h2



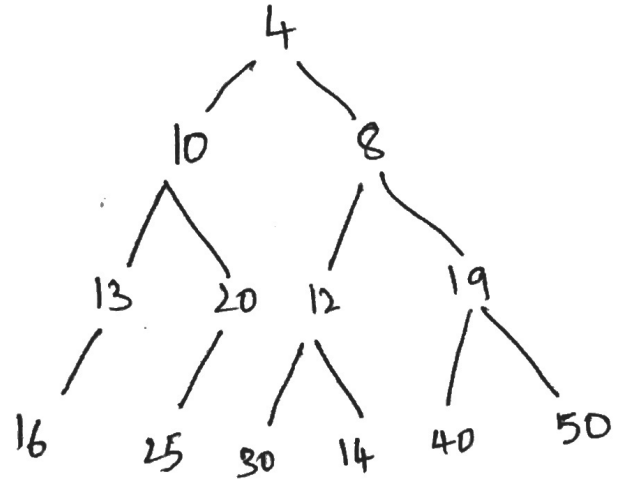
h2 has smaller root, so merge h1 with right subtree of h2



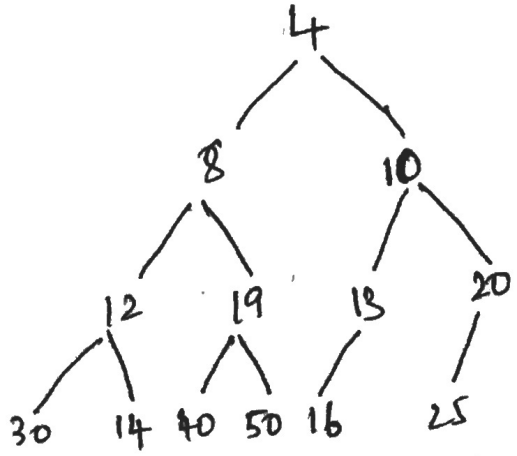
19 becomes right subtree of 8



8 becomes right subtree of 4



subtree swap due to leftist violation

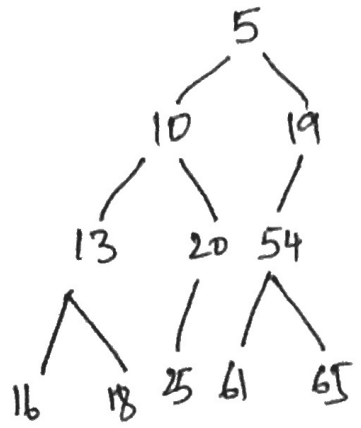


merge complete

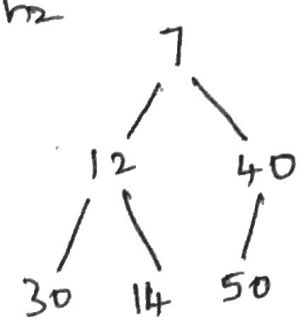
(6)

(4)

h_1

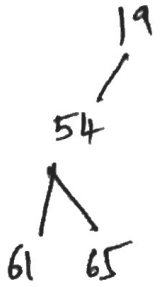


h_2

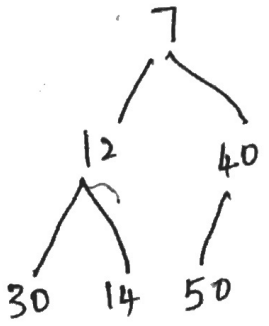


h_1 has smaller root, so merge h_2 with right subtree of h_1

h_1



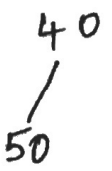
h_2



h_2 has smaller root, so merge h_1 with right subtree of h_2

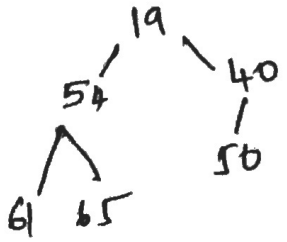
h_1

h_2



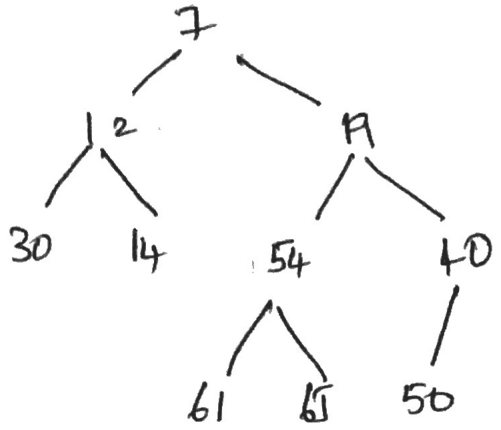
h_1 has smaller root, so merge h_2 with right subtree of h_1

h_1

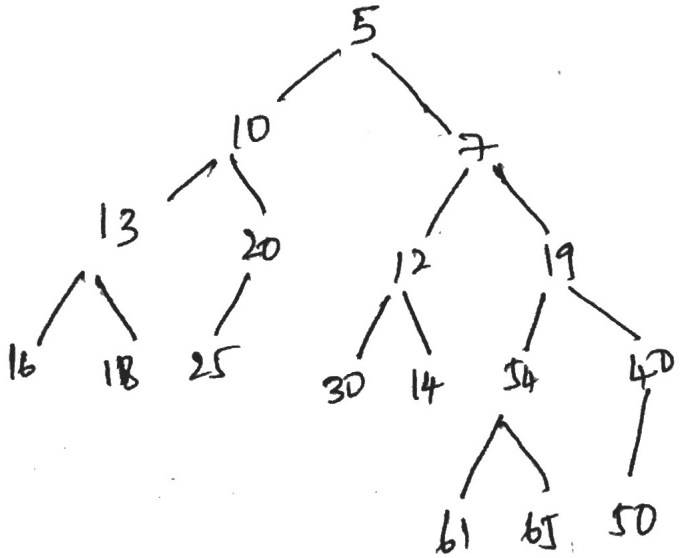


19 becomes the right subtree of 7

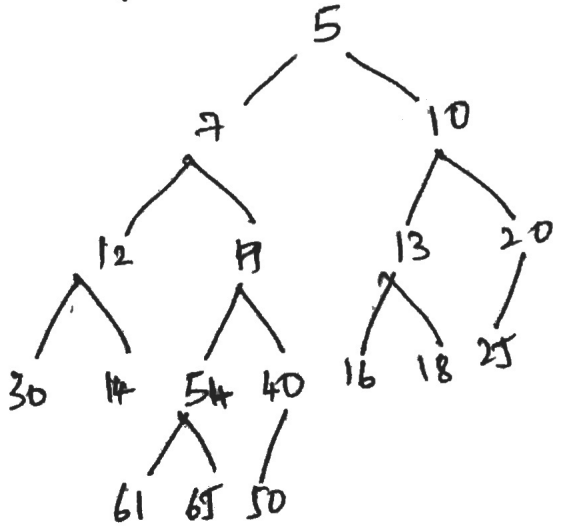
h2



7 becomes the right subtree of 5



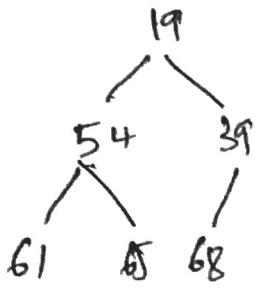
subtree swap due to leftist violation.



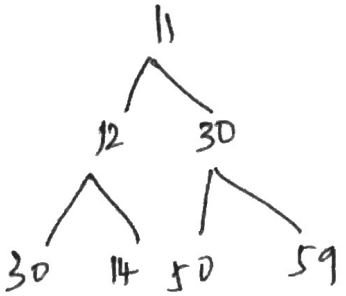
(7)

(5)

h1

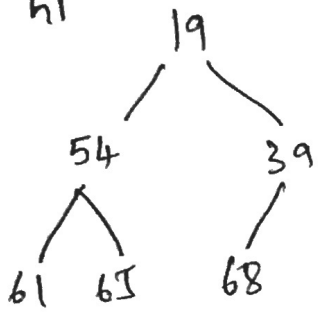


h2



h2 has smaller root, so merge h1 with right subtree of h2

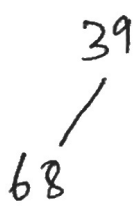
h1



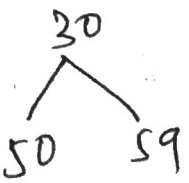
h2

h1 has smaller root, so merge h2 with right subtree of h1

h1

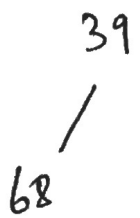


h2



h2 has smaller root, so merge h1 with right subtree of h2

h1



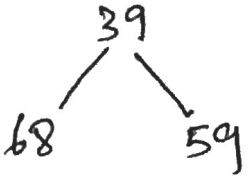
h2

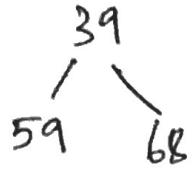
59

h1 has smaller root, so merge h2 with right subtree of h1

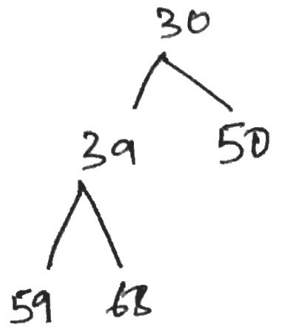
merge & swap subtrees

h1



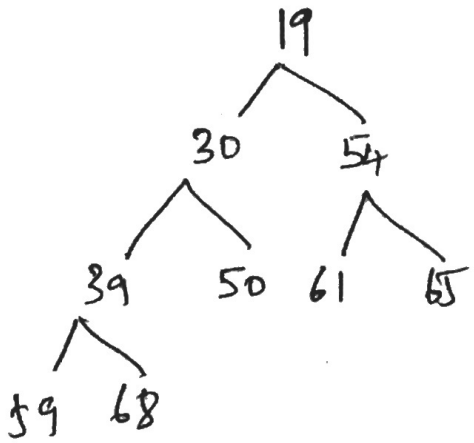


39 becomes the right subtree of 30

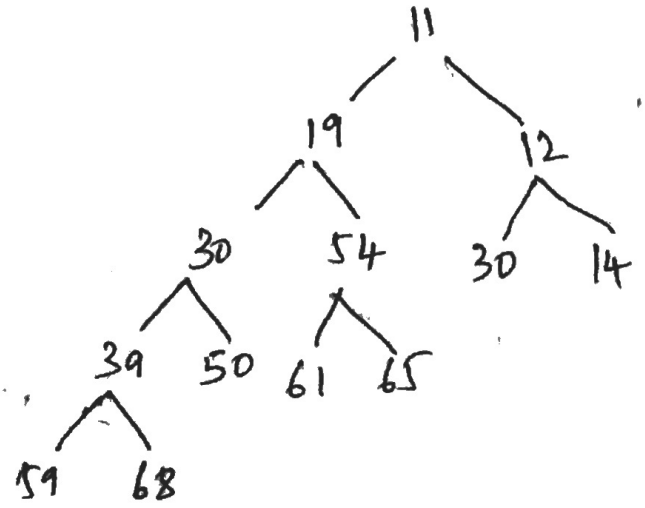


(swap)

30 becomes the right subtree of 19

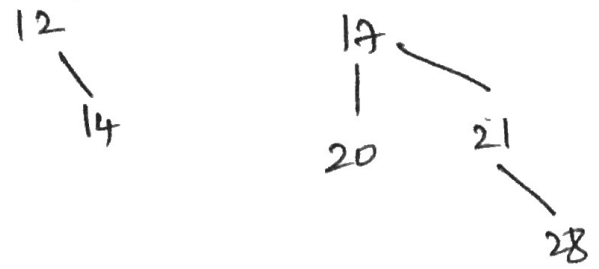


19 becomes the right subtree of 11 and swap

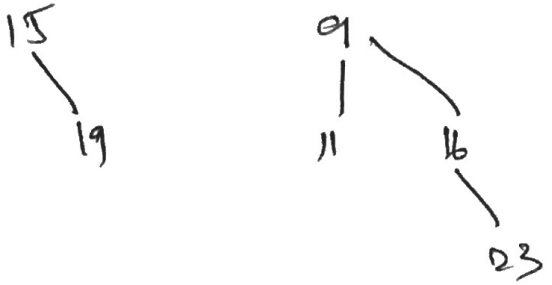


(8) (6)

h_1: 8

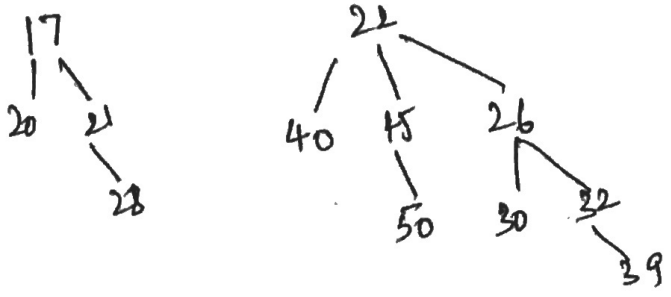


h_2: 6

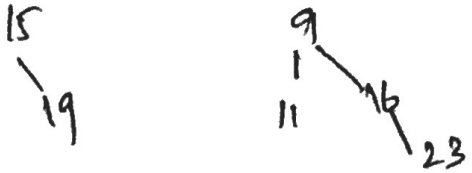


h_1 has 7 nodes and h_2 has 7 nodes.
Therefore the final queue has 14 nodes.
So the final queue has 8, 4 and 2 nodes trees.

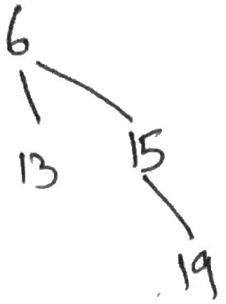
(9) h_1: 13

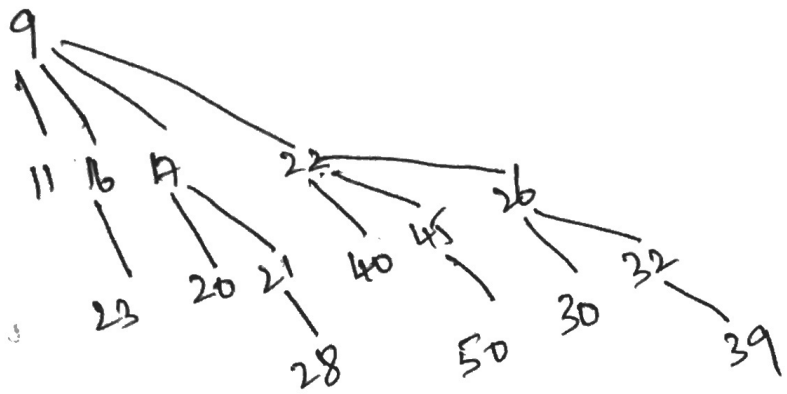


h_2: 6

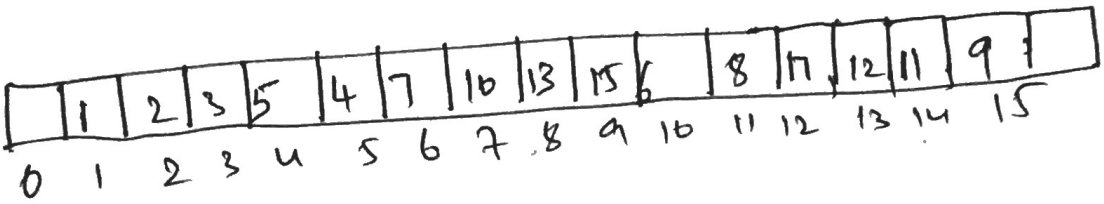


h_1 has 13 nodes and h_2 has 7 nodes.
Therefore, the final queue has 16 and 4 node trees.





(10) (a)



(b) If parent's index is i, then

→ left child's index is $3i-1$

→ middle child's index is $3i$

→ right child's index is $3i+1$.

(10) (e) If child's index is i, then

$$\text{parent's index} = \begin{cases} \lceil i/3 \rceil & \text{if } i \bmod 3 = 2 \quad (\text{ceil}) \\ i/3 & \text{if } i \bmod 3 = 0 \\ \lfloor i/3 \rfloor & \text{if } i \bmod 3 = 1 \quad (\text{floor}) \end{cases}$$