HW-5

Neel Katrodiya 1002254987

Q6.1-1 Min and Max no. of elements in heap of height h.

minimum number of elements:

let $h=0$, $n=1$

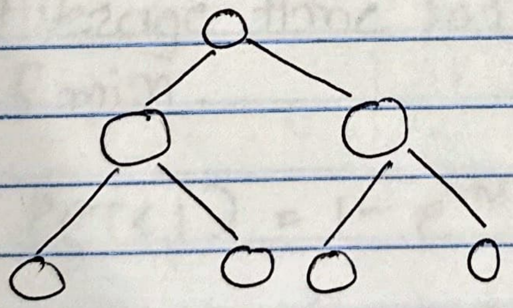
○

	min
$h=0$	$n=1$
$h=1$	$n=2$
$h=2$	$n=4$
$h=3$	$n=8$
$h=n$	$n=2^n$

min. no of elements $\boxed{= 2^h}$

Maximum no. of elements:

let height $(h)=2$, number $(n)=7$



	max	
$h=0$	$n=1$	$=1+(1-1)$
$h=1$	$n=3$	$=2+(2-1)$
$h=2$	$n=7$	$=4+(4-1)$
$h=3$	$n=15$	$=8+(8-1)$
$h=n$	$n=2^n+(2^n-1)$	

max no. of elements $\boxed{= 2^{n+1}-1}$

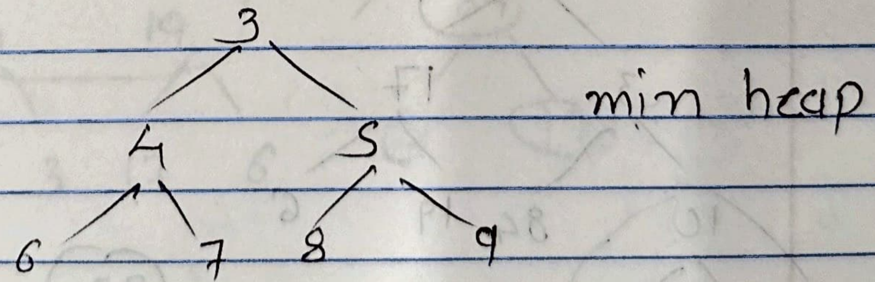
Q-6.14 In a max-heap, the smallest element would reside in one of leaf node, This is because of all nodes and the smallest element is distinct from all other elements in the heap.!

- Leaf node have no children, so they are not restricted by the heap property relative to descendants.

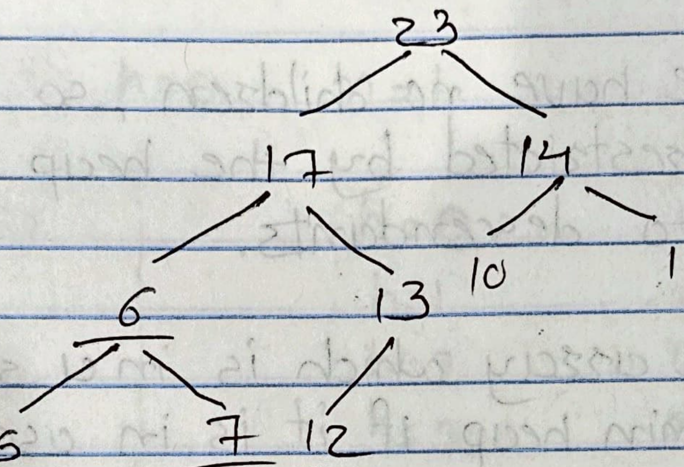
Q-6.15 Yes, the array which is in a sorted order a min heap if it is in ascending order.

Example

3, 4, 5, 6, 7, 8, 9



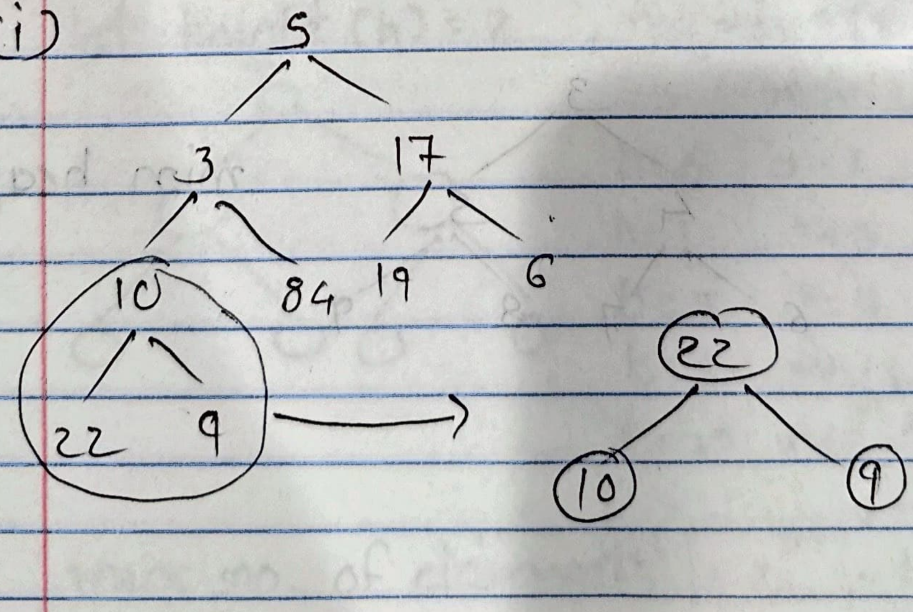
Q-6.1-6 No, the array with values ⟨23, 17, 14, 6, 13, 10, 1, 5, 7, 12⟩ is not a max heap because 6 has right child 7 which is greater than or equal to 6.



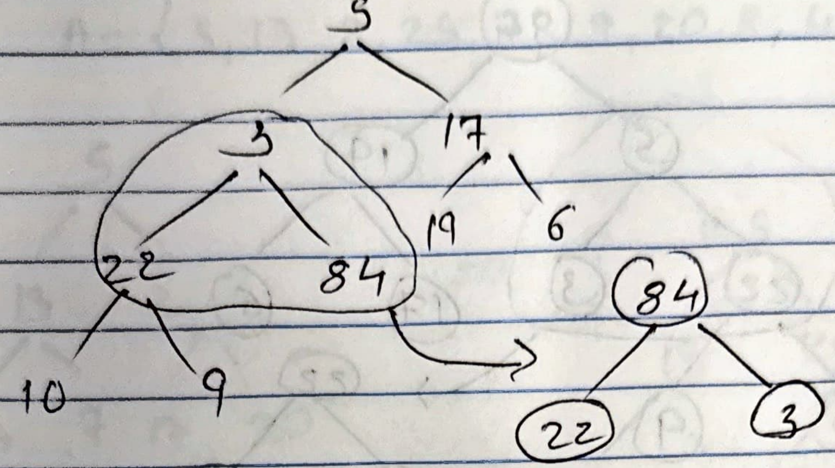
greater than parent.

Q-6.3-1 Array A = ⟨5, 3, 17, 10, 84, 19, 6, 22, 9⟩

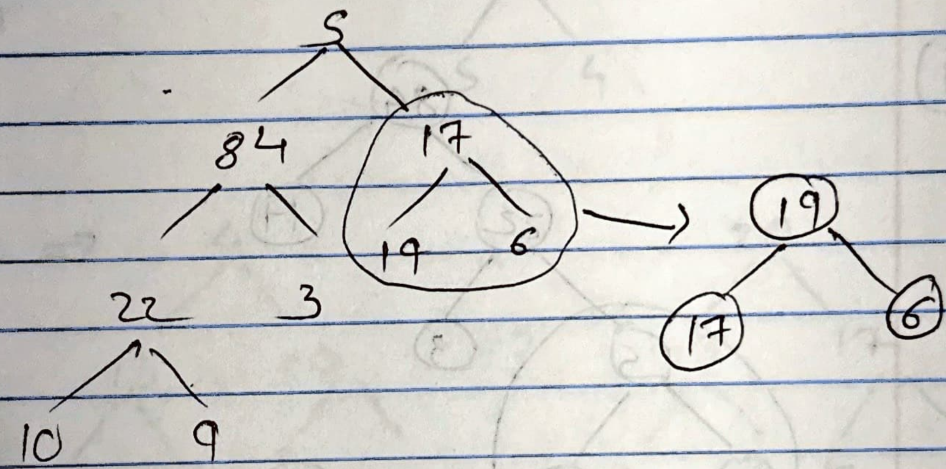
(i)



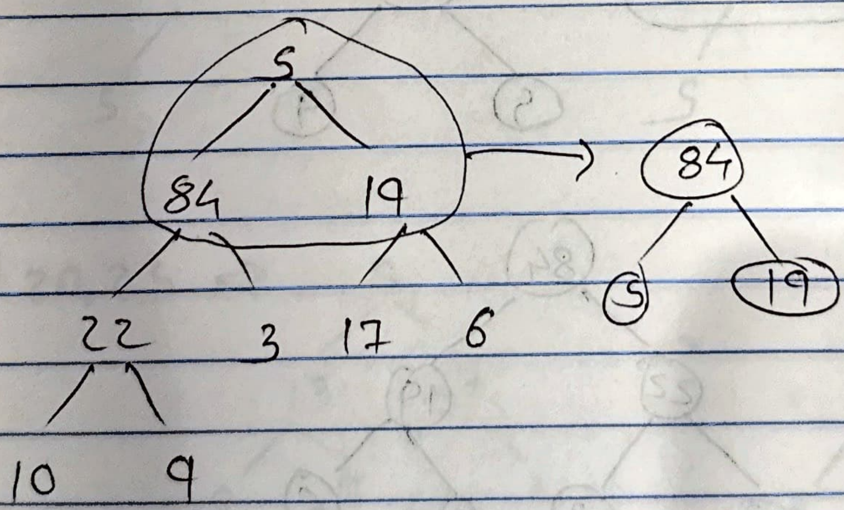
ii)


5
3
17
19
6
22
84
10
9
84
22
3

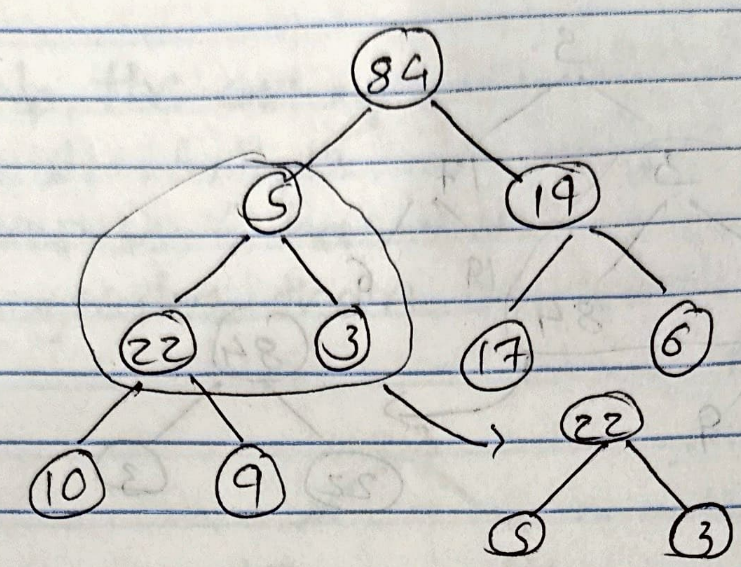
iii)


5
84
17
19
6
22
3
10
9
19
17
6

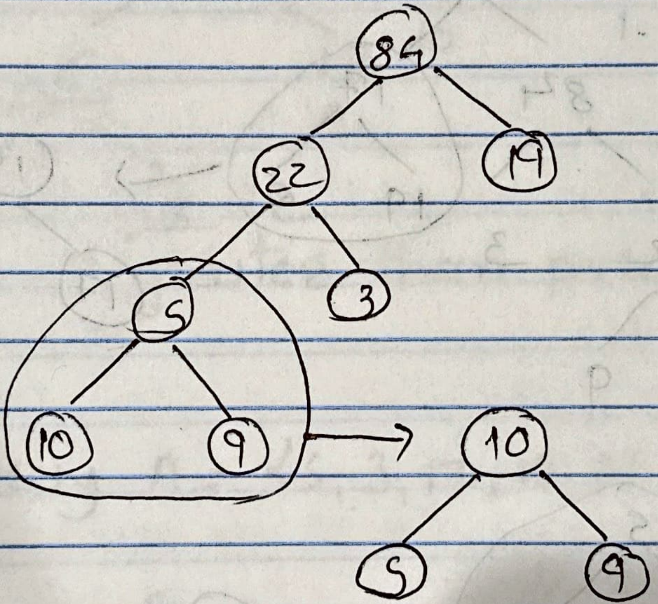
iv)


5
84
19
22
3
17
6
10
9
84
5
19

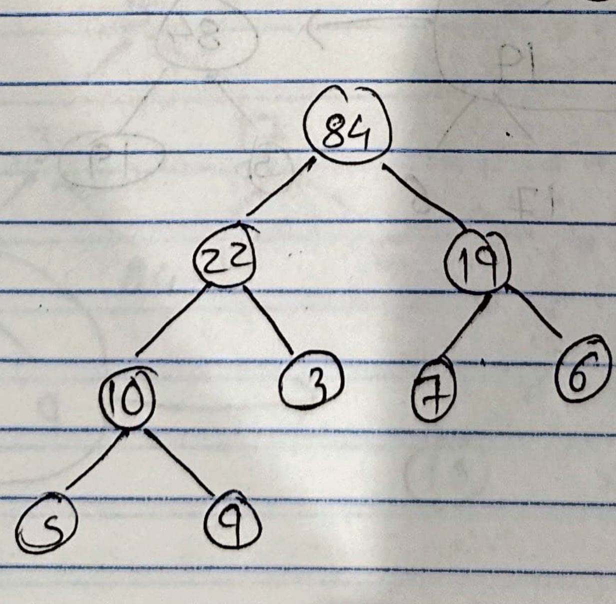
v)



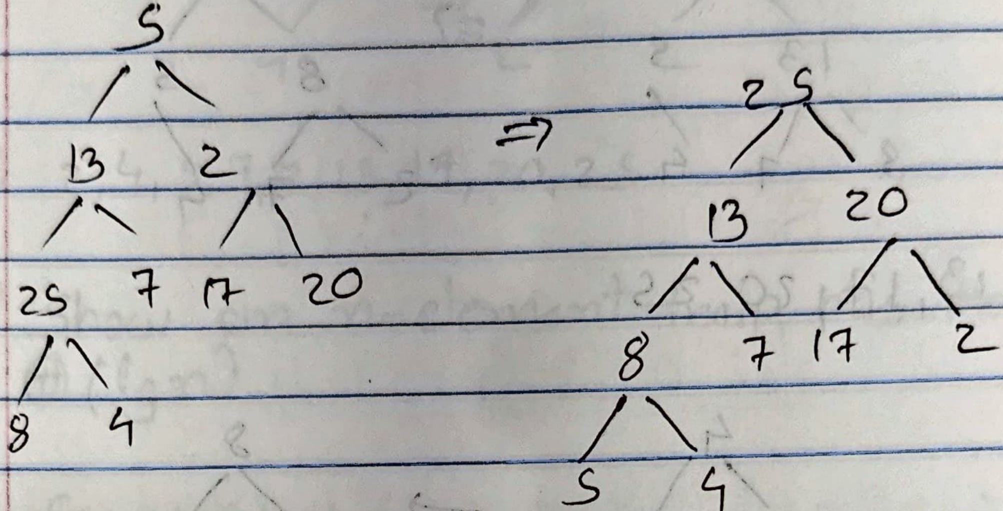
vi)



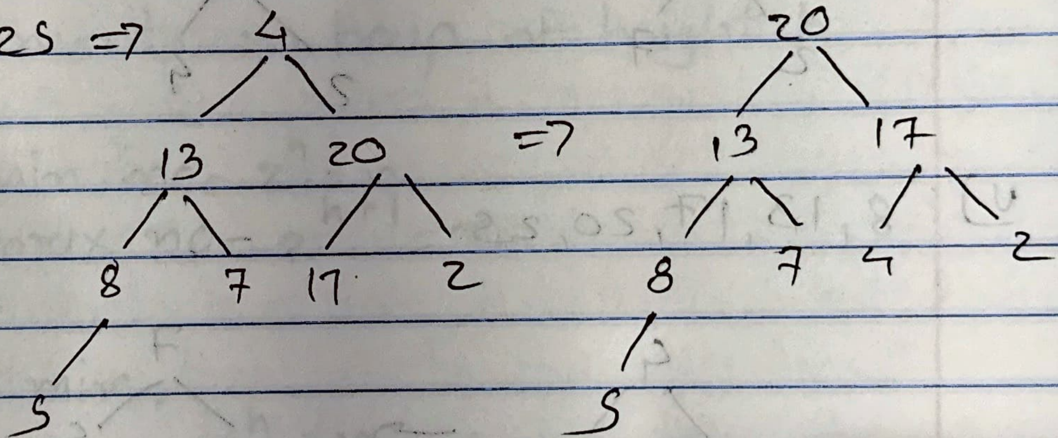
vii)



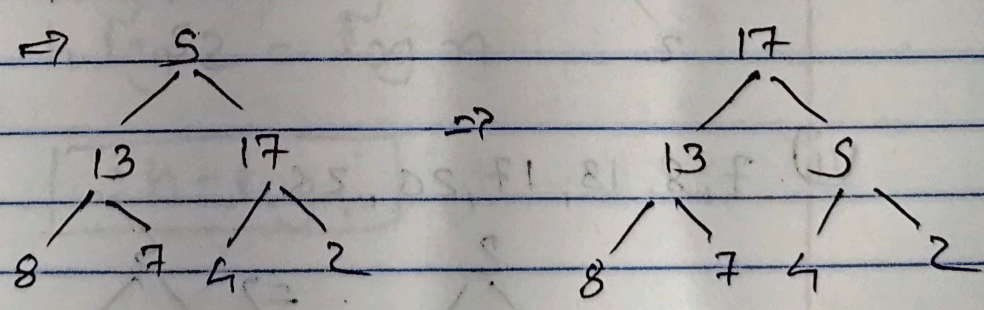
Ex-6.4.1 $A = \{5, 13, 2, 25, 7, 17, 20, 8, 4\}$



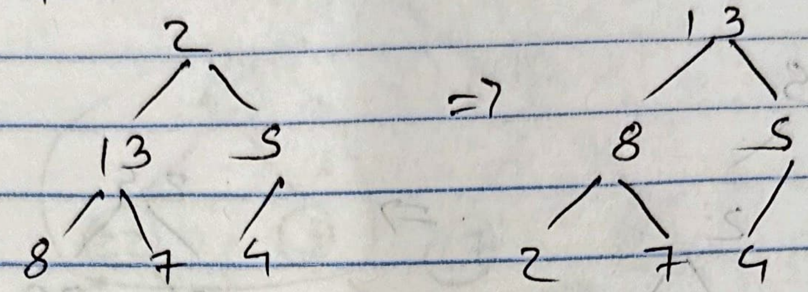
(i) 25 $\Rightarrow$



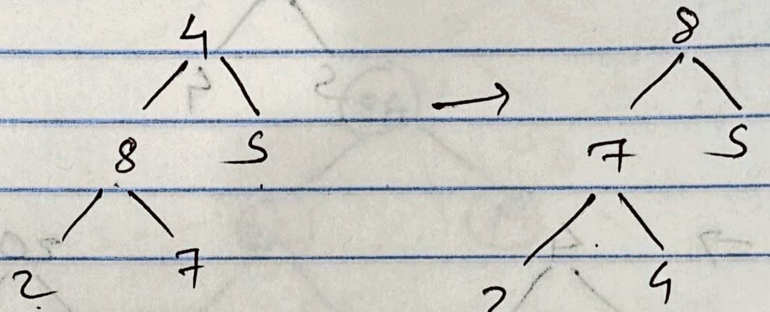
(ii) 20, 25 $\Rightarrow$



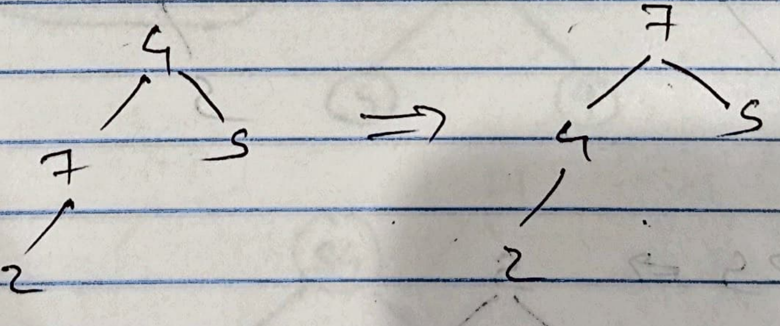
iii) 17, 20, 25



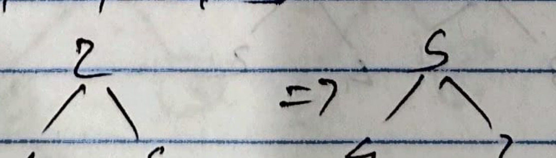
iv) 13, 17, 20, 25



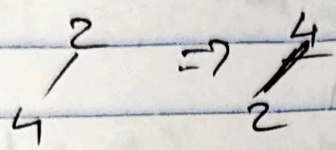
v) 8, 13, 17, 20, 25



vi) 7, 8, 13, 17, 20, 25



vii) 5, 7, 8, 13, 17, 20, 25



viii) 2, 4, 5, 7, 8, 13, 17, 20, 25

17) Show an n-element heap has height $\Theta(\lg n)$

From 6.1-1
we know that max & min number of elements in heap of height h

min no $= 2^h = n$
max no $= 2^{h+1} = n$

at min,

$$2^h = n$$
$$\log 2^h = \log n$$
$$h \log 2 = \log n$$

$$\boxed{\therefore h = \log n}$$

at max,

$$2^{h+1} - 1 = n$$
$$2^{h+1} = n+1$$
$$\therefore \log 2^{h+1} = \log(n+1)$$
$$\therefore (h+1)\log 2 = \log(n+1)$$
$$\boxed{\therefore h = (\log(n+1)) - 1}$$

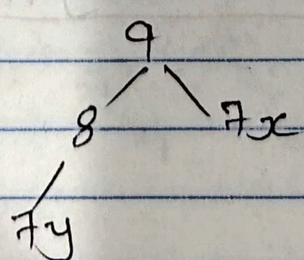
we showed that n element heights have log n.

2) a) Is heap sort a stable sorting Algorithm?

No, Operations in heap sort can change the relative order of equivalent keys. Therefore, heapsort is not stable sorting algorithm.

$(9, 8, 7x, 7y)$

1. $(9, 8, 7x, 7y)$



2. (7y, 8, 7x | 9) $\xrightarrow[\text{heaping}]{}$ (8, 7y, 7x | 9)

3. (7x 7y | 8, 9) $\xrightarrow[\text{heaping}]{}$ (7x, 7y / 8, 9)

4. (7y | 7x, 8, 9) => (7y, 7x, 8, 9)

swapped

if it was stable it would be (7x, 7y, 8, 9)

(B) Is Bubble sort a stable sorting algorithm?

→ Yes, bubble sort is stable sorting algorithm because it maintains relative order of elements with equal values after sorting.

c) Why might it be important to use a stable sorting algorithm?

=> A stable sorting algorithm maintains relative order of items with equal sort keys.

In situations when your data ~~is~~ b
is already in order to it by another
sort key.

For example, you have rows in spread-
sheet containing students data by
default sorted by name.

You would also like to sort it by
gender while remaining sorted
order of names.