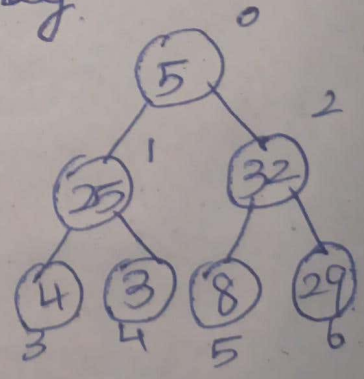
Heap sort:-

Algorithm:-

1. Rearrange the elements so that they form max heap.
2. Repeat until heap has one element.
   - 2.1 swap root with last element
   - 2.2. Remove last element of the heap.
   - 2.3 heapify remaining elements of the heap.
3. finally we get sorted array.

Example:-

5	25	32	4	3	8	29
0	1	2	3	4	5	6



arraysize = 7.

1. Build max heap.

Start node to heapify,

① = n//2

= 7//2 - 1

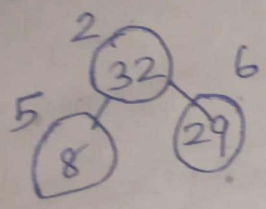
= 3 - 1

= 2

[Leaf nodes = non leaf node + 1]

find first non leaf node.

total element - 1



$l = 2*i + 1$

$= 2(2) + 1$

$l = 5$

$r = 2*i + 2$

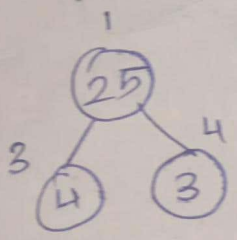
$= 2(2) + 2$

$= 4 + 2$

$r = 6$

32 is larger no swap.

②



$l = 2*i + 1$

$= 2(1) + 1$

$= 2 + 1$

$l = 3$

$r = 2*i + 2$

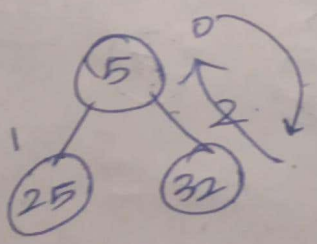
$= 2*(1) + 2$

$= 2 + 2$

$r = 4$

25 is larger so no swap

③



$l = 2*i + 1$

$= 2(0) + 1$

$= 1$

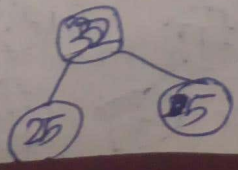
$r = 2*i + 2$

$= 2(0) + 2$

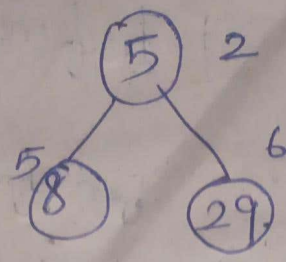
$= 2$

largest = 32

have to swap



here have to heapify

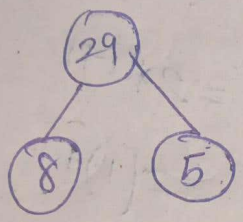


$l = 2i+1$ $\quad r = 2i+2$

$l = 5$ $\quad r = 6$

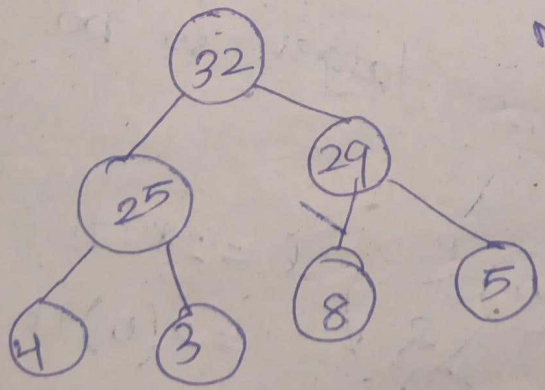
largest = 29.

have to swap



now the final heap will be,

max heap.

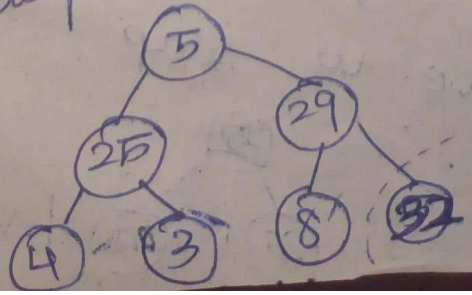


Step 2:- (I)

(1) swap root with last element

(II)

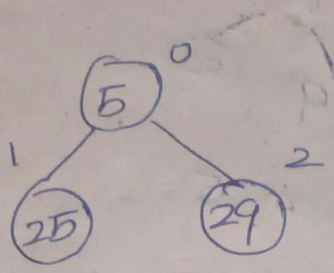
(2) remove last element



(3) Heapify remaining element.

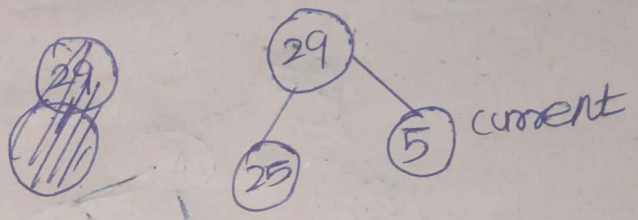
Array size = 6

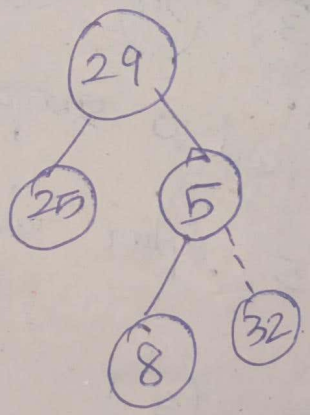
current element = 0



largest = 29

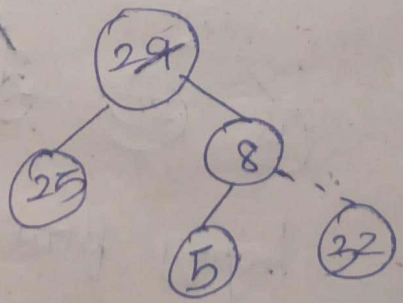
swap needed





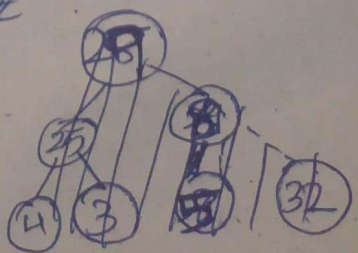
largest = 8

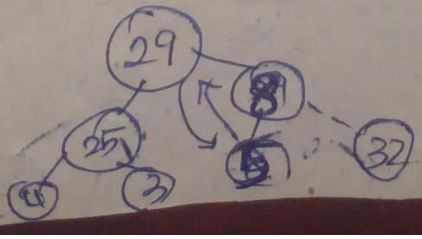
have to swap

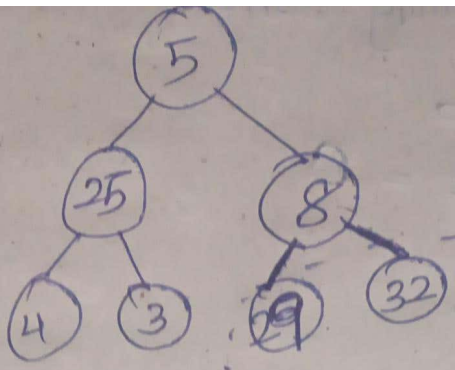


II

1 Swap root with last element

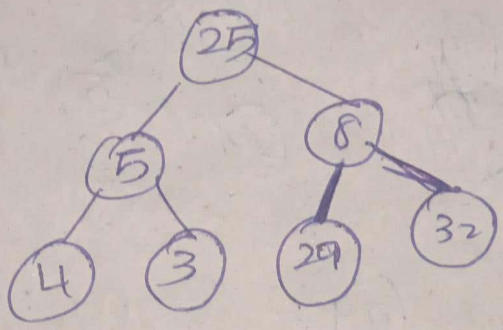






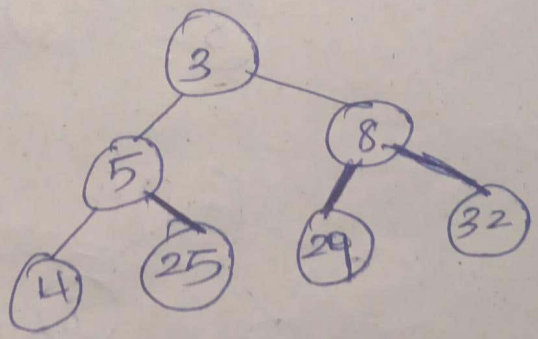
Heapify from 5.

Largest = 25.



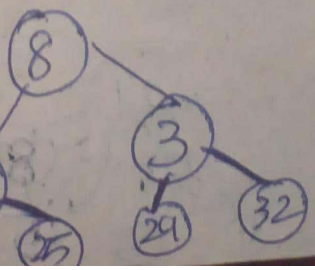
No need to swap.

(III) pop 25. replace with 3

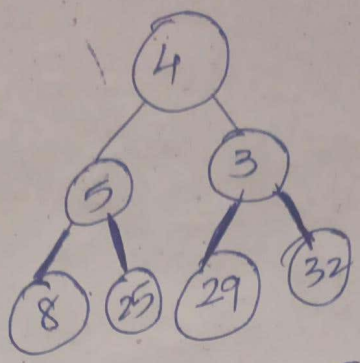


Heapify from 3

Largest = 8.

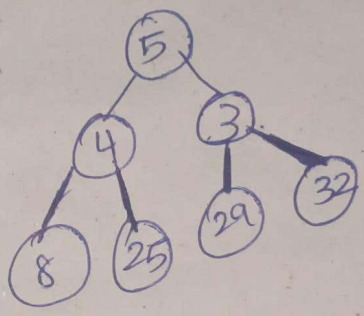


IV

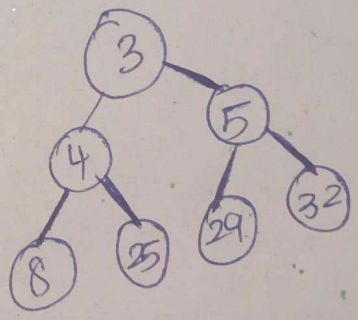


7

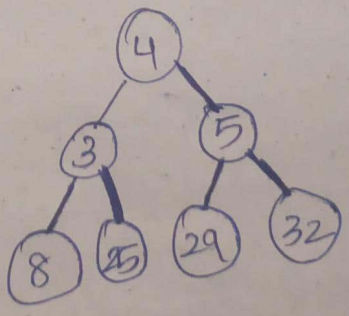
largest = 5

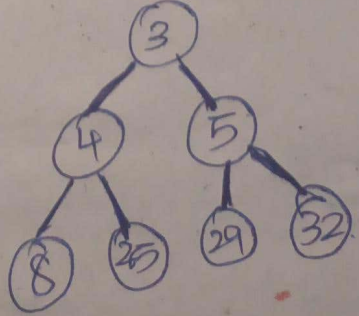


V



VI





Sorted array : 3, 4, 5, 8, 25, 29, 32