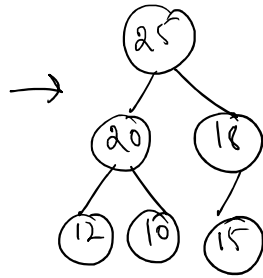
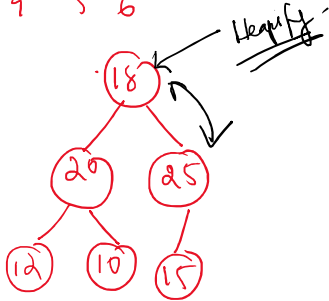
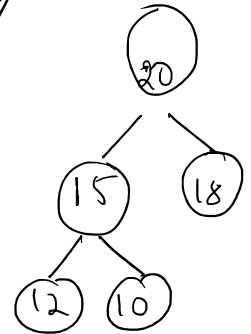
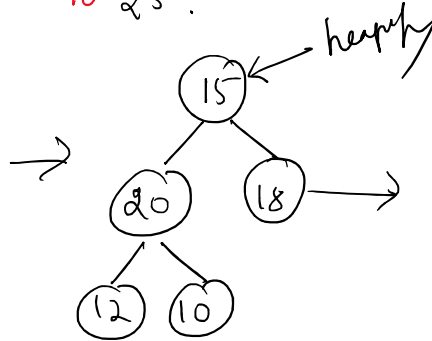


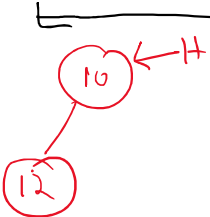
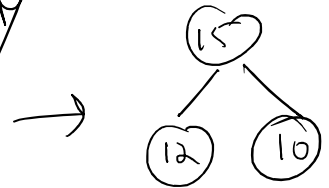
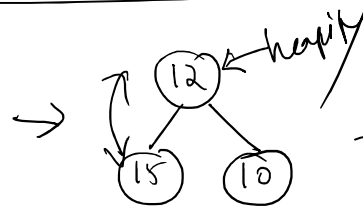
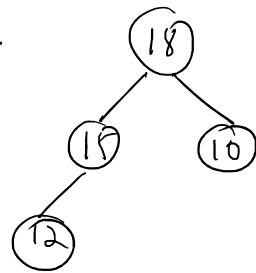
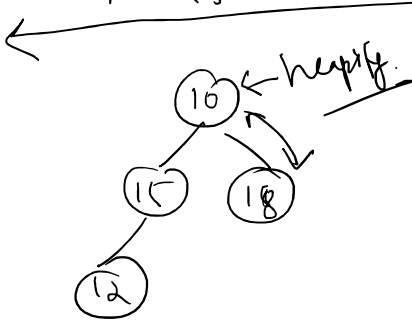
- ① Root value को अलग से लिखो।
- ② Root value = Last node value.
- ③ Last node delete
- ④ Heapify (Root).



40 25.



40 25 20 18 15 12 10



$TC \rightarrow O(n \log n)$ $SC \rightarrow O(1)$.

Linear Sorting Algo →

Counting Sort - Count frequency of elements.

2 1 3 3 1 2 2 1.

① counter $\xrightarrow{\text{size}}$ $\max(arr) + 1$ k_z

$TC \rightarrow O(n+k)$
 $SC \rightarrow O(k)$

CountArr

0	2	3	2
0	1	2	3

$\swarrow \quad \searrow$
 1 1 1 2 2 2 3 3

→ Disadv

① -ive integer.

② 10^{18}

TC $\rightarrow O()$

RADIX Sort -

\downarrow 1's place
 904
 046
 005
 074
 062
 001

Linear
 10's place
 001
 062
 904
 074
 005
 046

→

100's place.

\downarrow
 001
 904
 005
 046
 062
 074
 001
 074

① Find max element = 904

② Find no. of digit in max ele = 3.

③ Add 0's in front of other numbers.

$TC \rightarrow O(n \times k)$

→ 10^{18}

→ 19 digit.

H/w - Shell Sort.