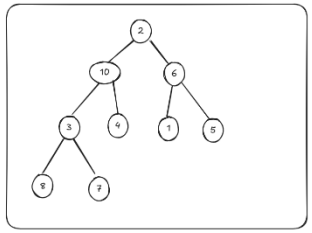
Sort in ascending order -> Use "Max heap" concept in heap sort

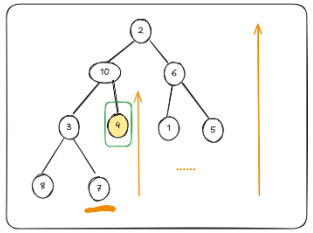
Unsorted



Heapify (n, i)

n is no. of elements
i ranges from (n/2)-1 to 0

Start from (n/2) - 1



Decrement i for each function call, until 0

index in Max Heapify keeps changing in every iteration

Max Heapify (n, index):

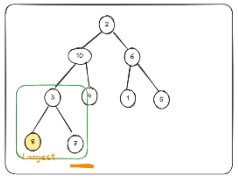
Start from index
Assign largest = index
Assign left = (2*index) + 1 -> left child
Assign right = (2*index) + 2 -> right child

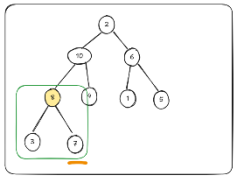
Make sure that the left < n, right < n

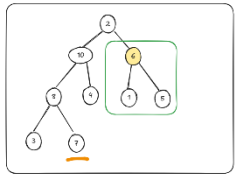
Swap the largest from left/right (if any) and index

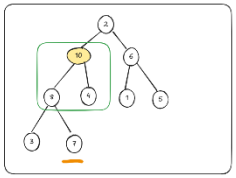
The previous largest is the new index

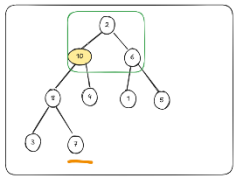
Repeat until largest and index are the same

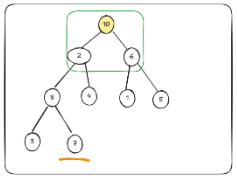

Longest

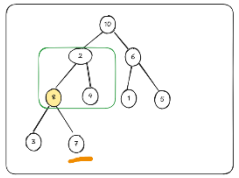


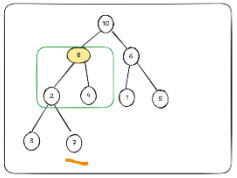


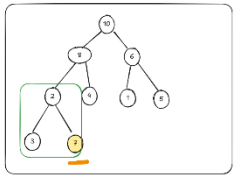


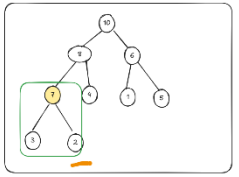








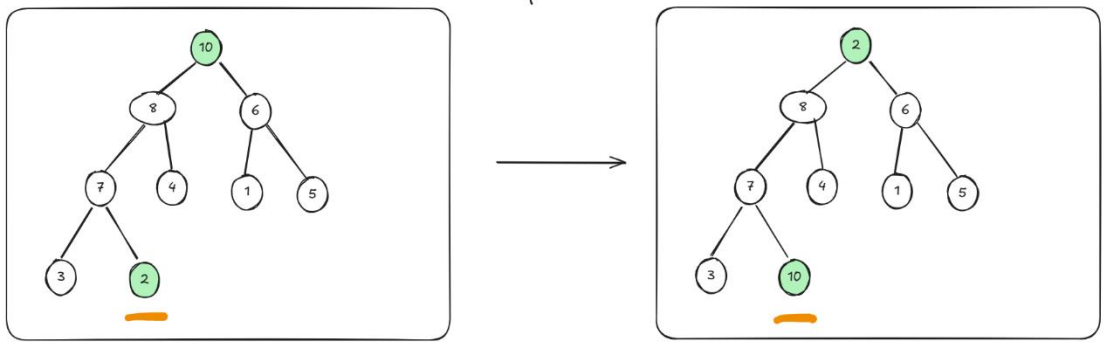




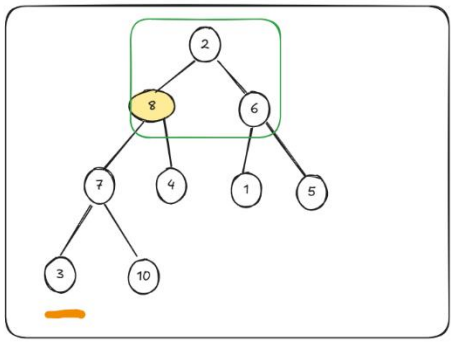
Swap i and 0 followed by Heapify (i,0)

where i ranges from n-1 to 1

Swap 0 and n



Heapify (n-1, 0)



In each function call,
start from 0, go until i < n-1

Decrement i for each function call, until 1

n in Max Heapify keeps changing in every iteration

Max Heapify (n, index):

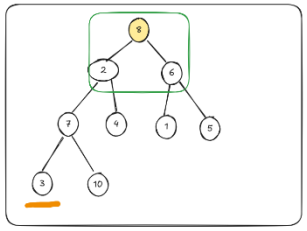
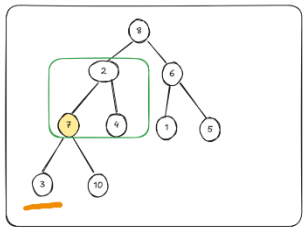
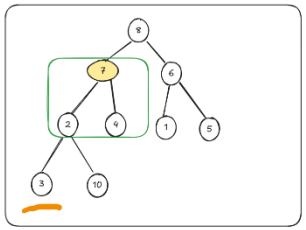
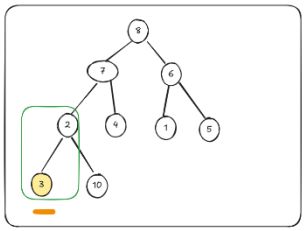
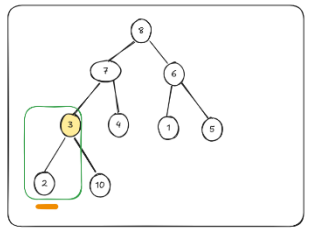
Start from index
Assign largest = index
Assign left = (2*index) + 1 -> left child
Assign right = (2*index) + 2 -> right child

Make sure that the left < n, right < n

Swap the largest from left/right (if any) and index

The previous largest is the new index

Repeat until largest and index are the same

Swap 0 and n-1