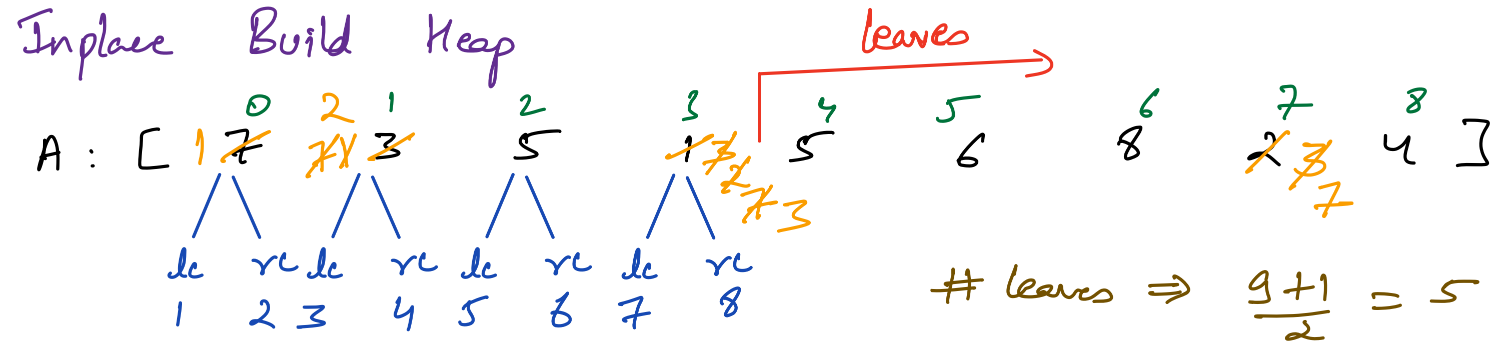
Welcome ☺

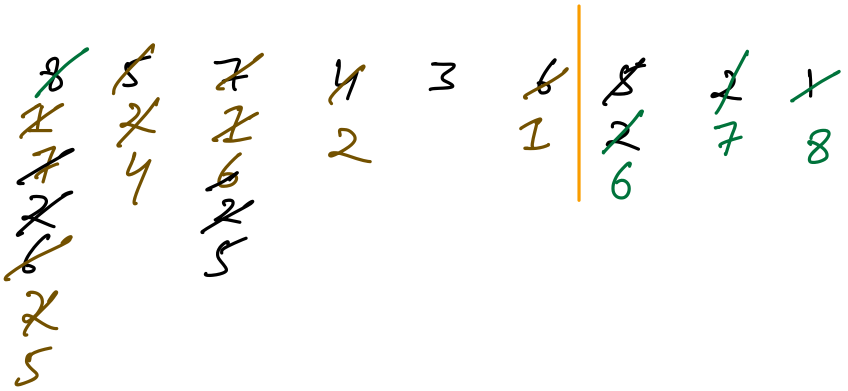
Agenda : Heap Sort
k^{th} largest ele.
Sort nearly sorted array
Median of stream of integers.

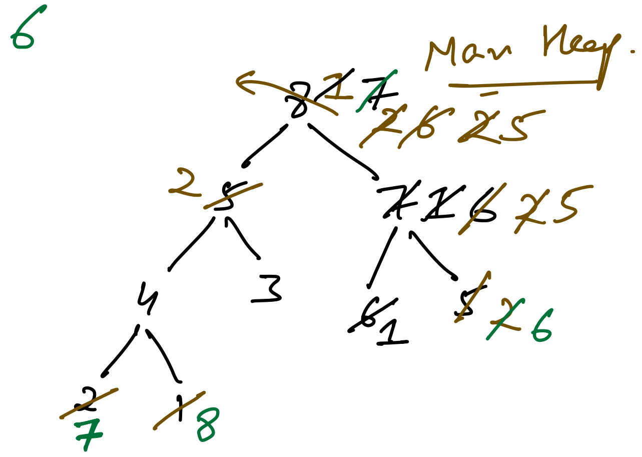
Heap Sort

Inplace Build Heap



leaves ⇒ $\frac{9+1}{2} = 5$





1) Build Max Heap

2) apply getMax( ) → N-1 times.
→ sorted data.

T.C → O(N + NlogN) ≃ O(NlogN)

S.C → O(1)

Stable sort ?

Heapsort is not stable ~~sort~~

Q Given an integer array, find K^{th} largest ele. ?

Soln 1) Sort the array & return $A[N-K]$

T.C $O(N\log N)$ S.C $O(1)$

2) Binary Search.

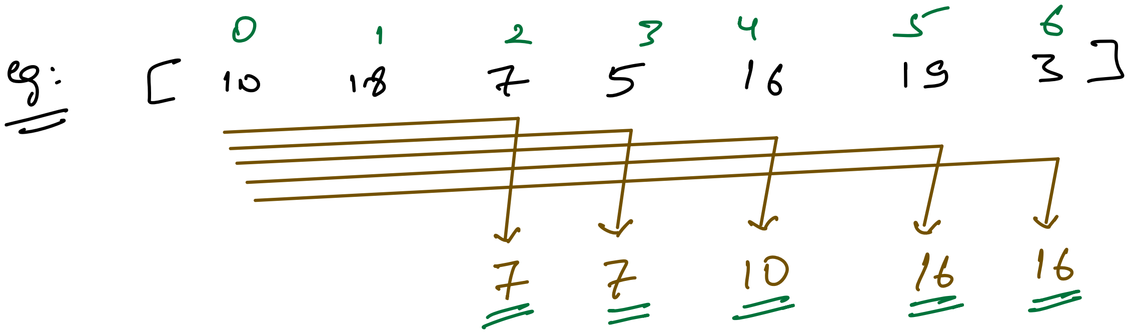
3) Heap Sort.

1) Build max Heap.

2) Call getMax K times.

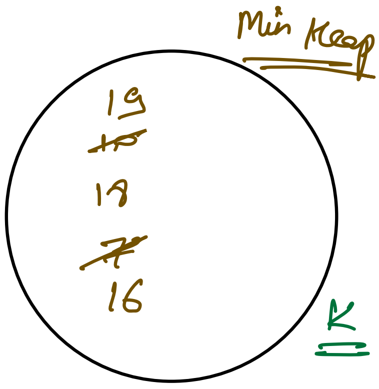
Q Given an integer array $\forall i \geq (K-1)$ find K^{th} largest ele from 0 to i

eg: [10 18 7 5 16 19 3] K=3



eg: [10 18 7]

3^{rd} Largest ⇒ 7 ⇒ Smallest element.



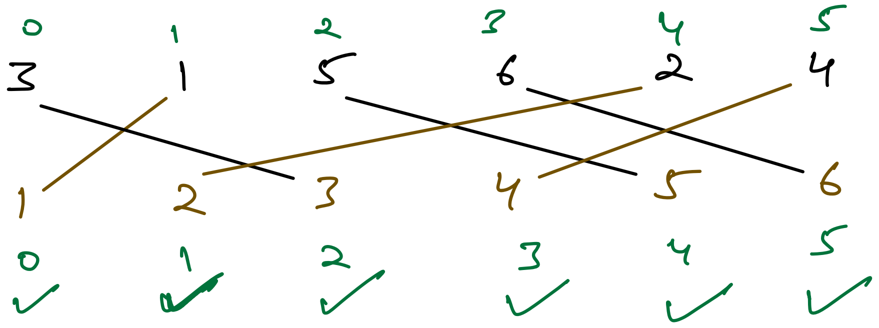
```
if ( A[i] ≤ root of min Heap ) → no update.
else {
        remove min element
        Insert ( A[i] )
}
```

T.C → $O(K + (N-K)\log K)$

S.C → $O(K)$

Q Given N elements in an array s.t every element is atmax K distance away from its pos^n in sorted order. | curr Index − idx in sorted array |
Sort the array.

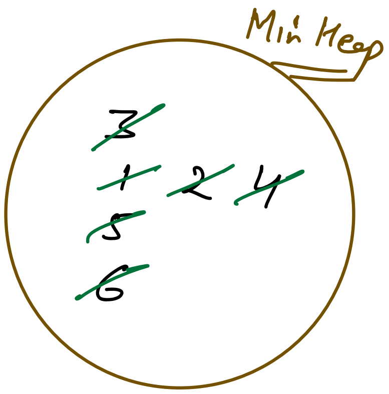
eg:



K=3

eg: 3 1 5 6 2 4

K=2



Code

```
// Build Min Heap.
 for( i -> 0 to K){
      h.insert(A[i])
 }
  ind=0
 for( i -> K+1 to N-1)
 {
      A[ind] = h.getMin()
      ind++
      h.insert( A[i])
 }

 while ( ! h.isEmpty())
 {
      A[ind] = h.getMin()
      ind++
 }
```

T.C → O(N log K)

S.C → O(K)

Q HackerRank

Given an integer input stream, find median at every step.

median ↓ middle element of sorted data.

i/p	9	8	7	3	6
o/p	9	8	8	7	7

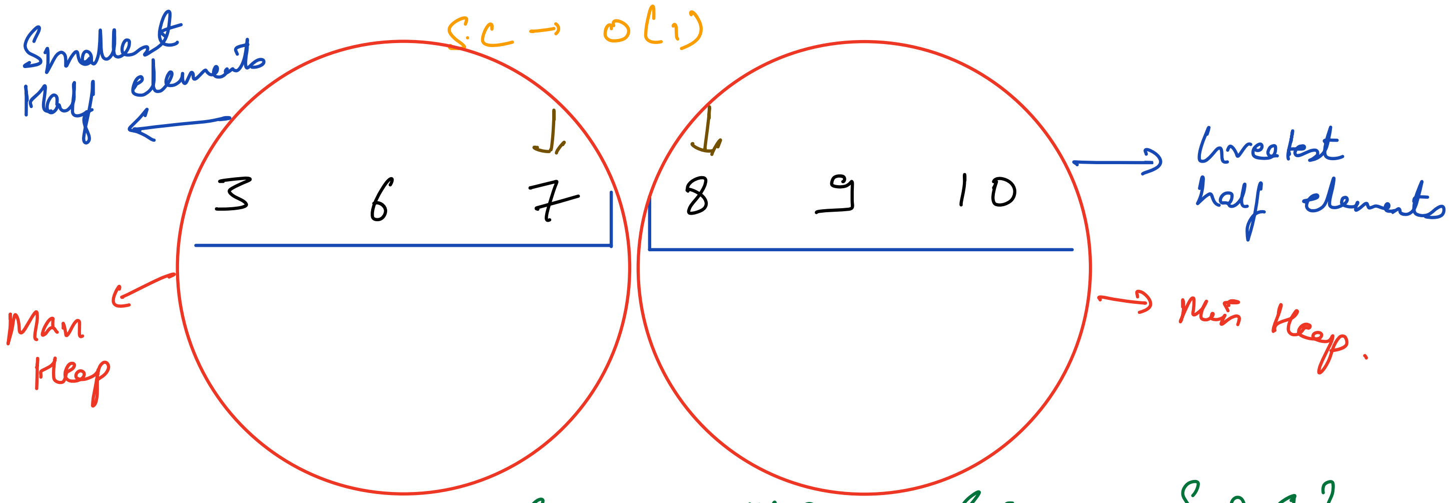
Soln

1) Brute force → sort at every stage & get middle.

↓ Insertion Sort.

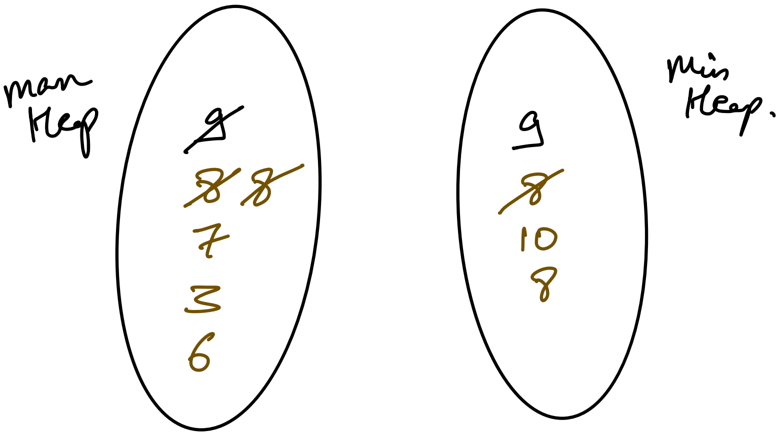
T.C → O(N^2)

S.C → O(1)



Max Heap Size — Min Heap Size = {0, 1}.

eg.	i/p →	9	8	7	3	10	6
		9	8	8	7	8	7



Code

```
∀ input $x$
if ( $x \le$ root of max Heap)
{
    insert ($x$) in max Heap.
    if( maxHeap size - minHeapSize > 1) // reshuffle
    {
        move root of maxHeap to min Heap.
    }
}
else {
    insert ($x$) in minHeap
    if( maxHeap size - minHeapSize < 0 ) // reshuffle
        move root of min Heap. to maxHeap
}
```

T.C → $O(N \log N)$

S.C → $O(N)$