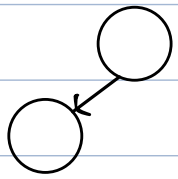
Q Given a complete Binary Tree with N nodes, find the number of leaf nodes.

N = 1

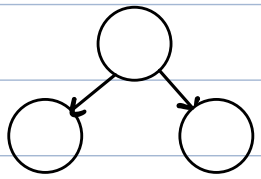
L = 1

N = 2



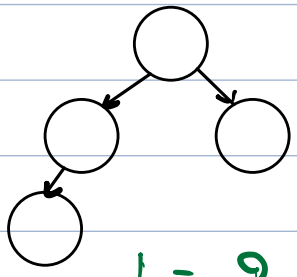
L = 1

N = 3



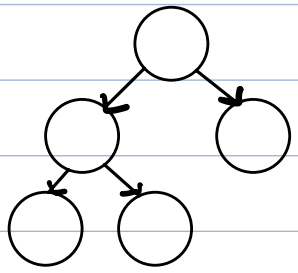
L = 2

N = 4



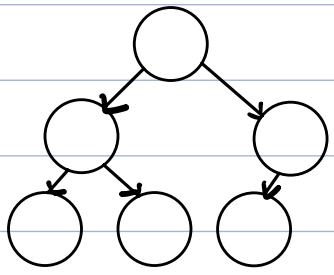
L = 2

N = 5



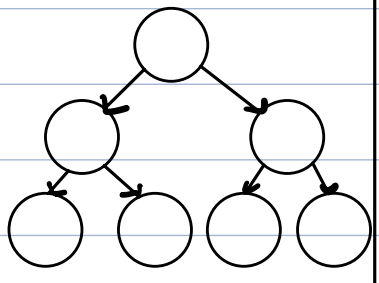
L = 3

N = 6



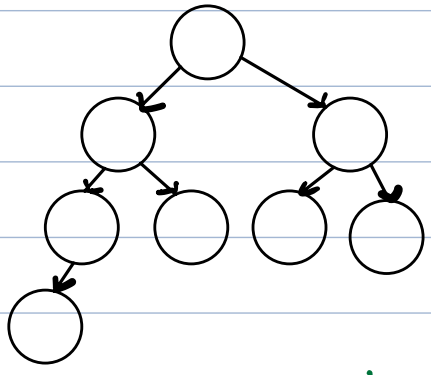
L = 3

N = 7



L = 4

N = 8



L = 4

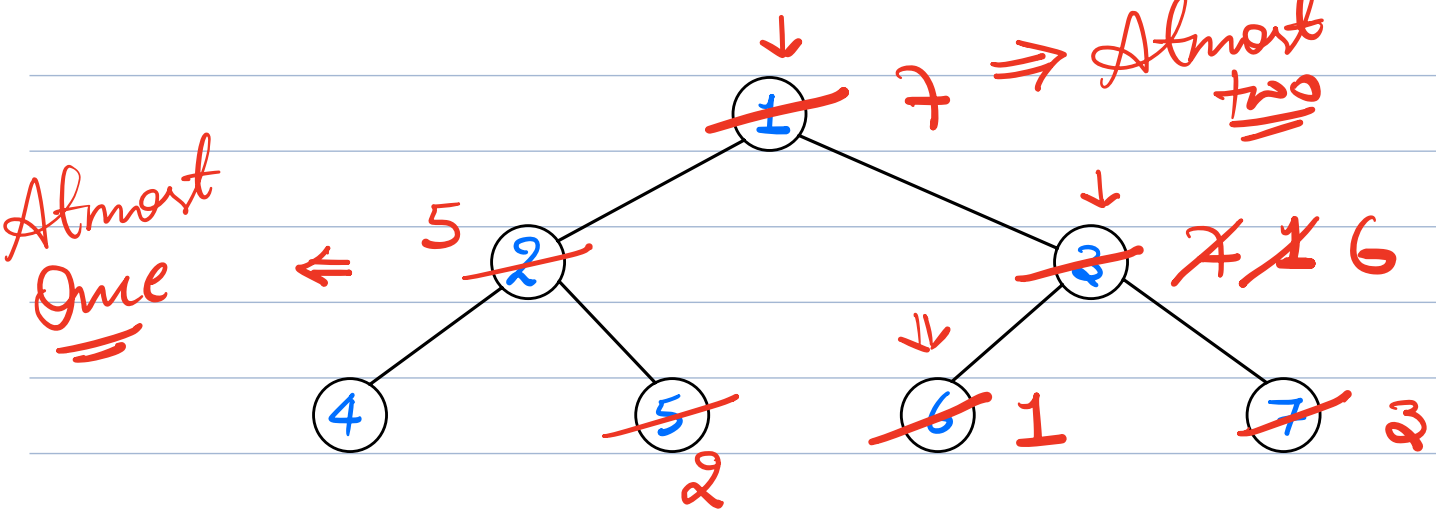
Leaf Nodes = $\left(\frac{N+1}{2}\right)$ = ceil $\left(N/2\right)$

Build Max Heap.

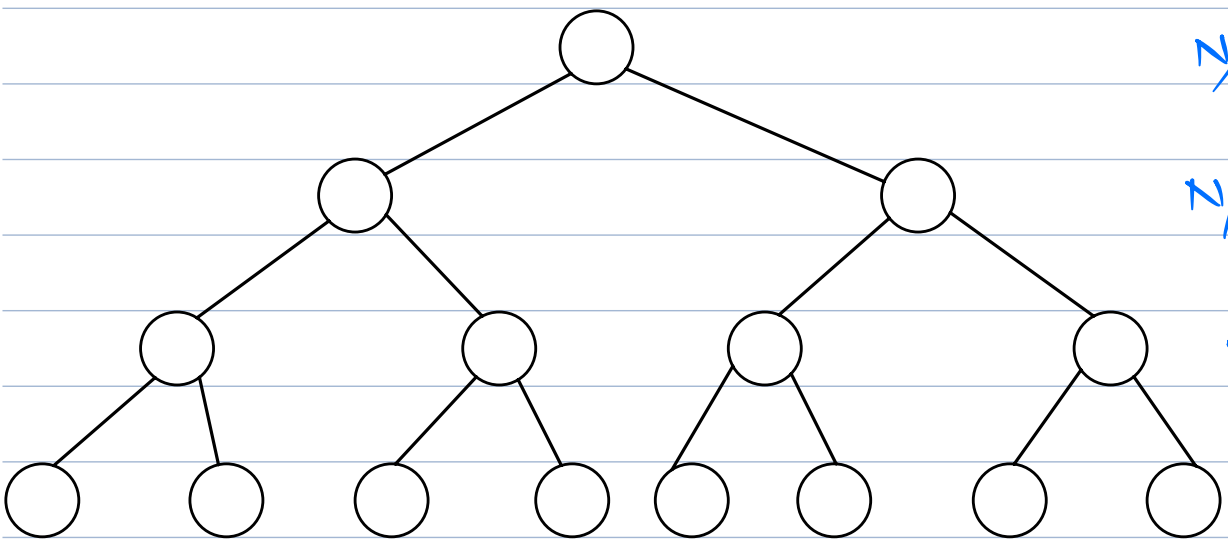
I/p : 1, 2, 3, 4, 5, 6, 7

N = 7

L = ??



leaf × # swaps



$N/16 \times 3$

$N/8 \times 2$

$N/4 \times 1$

$N/2 \times 0$

Total no. of swaps $= \left(0 \times \frac{N}{2}\right) + \left(1 \times \frac{N}{4}\right) + \left(2 \times \frac{N}{8}\right) + \left(3 \times \frac{N}{16}\right) \ldots\ldots$

$$= \sum_{i=0}^{\infty} i \times \frac{N}{2^{i+1}}$$

$$= \frac{N}{2} \times \left(\sum_{i=0}^{\infty} \frac{i}{2^i} \right)$$

$\left(\sum_{i=0}^{\infty} \frac{i}{2^i} \right) = 2$

Max no. of Swaps possible $= \frac{N}{2} \times 2 = N$

T.C. = $O(N)$
S.C. = $O(1)$

Pseudo Code

A ⇒ Input

```
for (i = N/2 ; i >= 0 ; i--) {
        downHeapify (A, i);
}
```

Heap Sort

Given an integer array of size N.
Sort the array using Heaps in-place. SC. O(1)

Idea

A = [. . . . .]

[1, 2, 4, 5, 6, 8, 9]

Min Heap → Sorted Array.

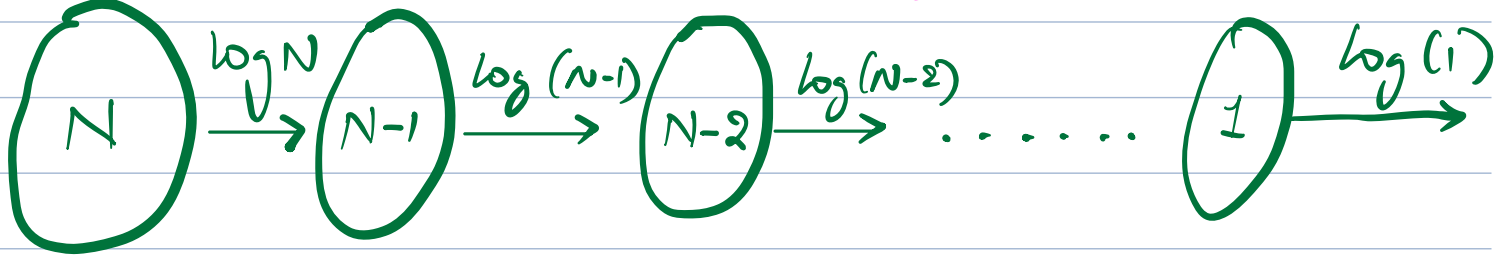
⇒ Extract min every time from the remaining elements to get the elements in sorted order.

Steps

1) Build a Min Heap ⇒ O(N)

2) Extract min element one by one & insert in a new array.

T.C. = O(N × log N) ✗



log(N) + log(N-1) + log(N-2) ..... log(1)

$= \log(N \times N-1 \times N-2 \ldots\ldots 1)$

$= \log(N!) \leq N \log(N)$

	0	1	2	3	4	5	6	7	8	9
A =	10	7	8	3	6	2	5	9	13	14

N times with a max Heap ⇒ Increasing array

Code

1) Build Heap → T.C. = O(N)

2)
```
j = N-1
while (j > 0) {
    swap(A[0], A[j]);
    j--;
    downHeapify(A, 0, (j+1));   // (j+1) = size
}
```

S.C. = O(1)

Given an integer array of size N. Find the K^{th} largest element.

Eg: $A = [8, 5, 1, 2, 4, 9, 7]$

$K = 3$

Ans = 7

Soln 1 ⇒ Sort the array in descending.

⇒ Return $A[K-1]$.

T.C. = $O(N \log N)$

S.C. = $O(1)$

Soln 2 Using Max Heap

1) Build a Max Heap

2) Delete Maximum K times.

T.C. = $O(N + K \log(N))$

Sol^{n} 3

Cricket Trials

⇓

Need 4 batsmen for the team.

10 eligible players for trial.

Every player plays 1 over

The 4 batsmen that scores most runs are selected.

12	10	9	8 ↓
Max	2nd max	3rd Max	4th Max.

12, 8, 9, 6, 10, 5, 11

↑

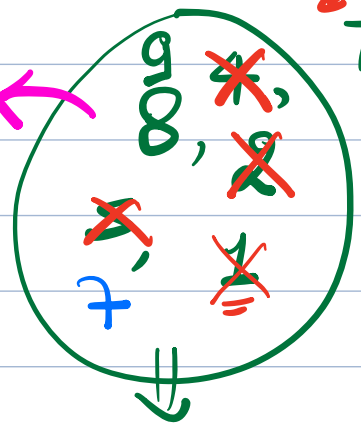
Should score more runs than the min runs scored by the top 4 scorer

K = 3 A = [8, 5, 1, 2, 4, 9, 7]

min

Top K elements

Max size Heap = K



→ Top k elements of the entire array.

Min Heap

Break 11:00PM

T.C. = O(N log(K))

S.C. = O(K)

Q) Given an integer array of size N find the Kth largest element for the elements from index 0 to i for ∀ i in [0, N-1]

Eg A = [10, 18, 7, 5, 16, 19, 3] K = 3

(0) (1) (2) (3) (4) (5) (6)

10 — x

18 — x

7 — 7

5 — 7

16 — 10

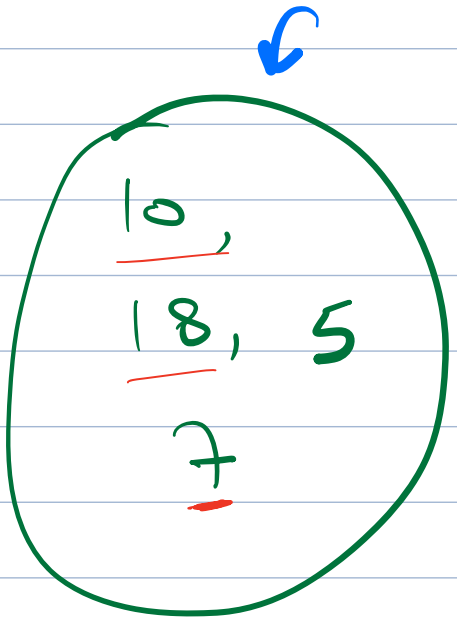
19 — 16

3 — 16

Sol^n 1) Max Heap

0	1	2	3	4	5	6
10	18	7	5	16	19	3

A = [10, 18, 7, 5, 16, 19, 3] K = 3



Max Heap

1) Extract Max K times.

2) Insert Max K elements back @ A[i+1]

2) Min Heap.

Code

```
MinHeap<int>  heap;
for(i=0 to K-1)
        heap.insert(A[i]);
```

```
}

ans[];
ans.add (heap[0]);

for (i=K; i<N; i++){

    if (A[i] > Heap[0]){

        deleteMin (Heap);
        insert (Heap, A[i]);

    }
    ans.add (Heap[0]);
}
```

Given an integer array which is **nearly sorted**, Sort the array.

Nearly Sorted ⇒ Every element is shifted from its correct sorted position by atmost K steps

	0	1	2	3	4	5	6	7	8
A =	13,	22,	31,	45,	11,	20,	48,	60,	50]

	0	1	2	3	4	5	6	7	8
Sorted A =	11	13	20	22	31	45	48	50	60]

Dist^{n} (4-0) (1-0) (5-2) (3-1) (4-2) (5-3) (6-6) (8-7) (8-7)

Sol^{n} 1)

Sol^{n} 2)

Given an infinite stream of integers. find the median of current set of integers.

1, 2, 5, 4, 3 .....

1, 1.5, 2, 3, 3 .....

[1, 2, 4, 3]

⇓

[1, 2, 3, 4] ⇒ Even

Average = 2.5

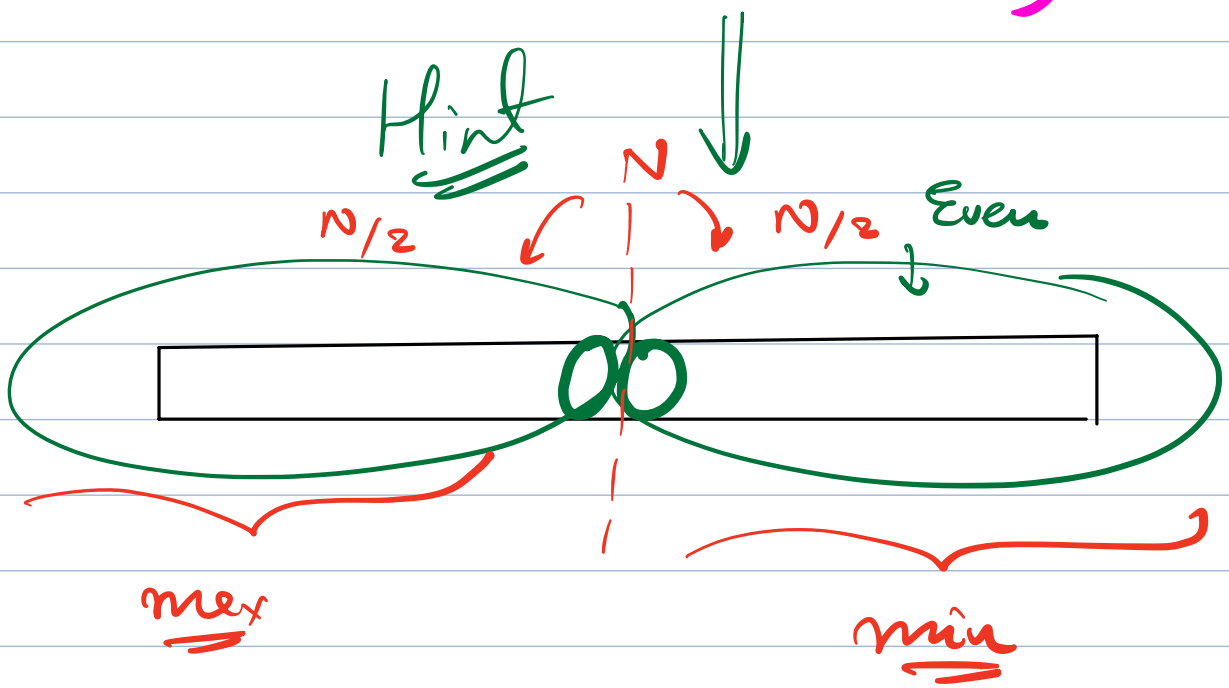
$A = [1, 2, 5, 4, 3, 6]$

find ans upto each $i \Rightarrow [0, i]$

Soln 1 Always maintain a sorted dataset (insertion Sort)

N elements $\Rightarrow$ T.C. $= O(N^2)$

Hint



H.W.

23rd (Next Friday) → Contest

⇓

Weekend

Monday ⇒ Greedy

← Wednesday → Heaps + Revision

+

Doubts on

Trees + Heaps +

Greedy

As many questions