

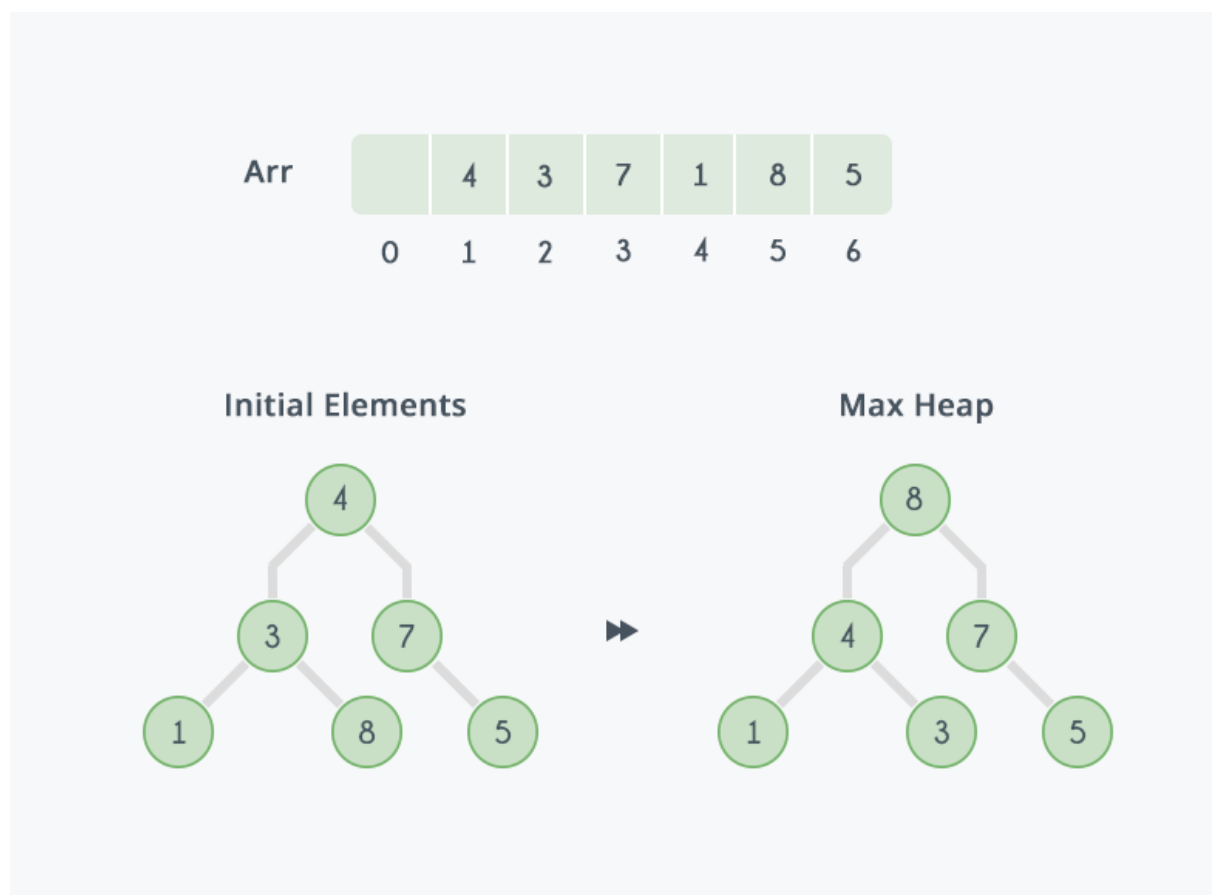
# HEAP SORT

Heaps can be used in sorting an array. In max-heaps, maximum element will always be at the root. Heap Sort uses this property of heap to sort the array.

Consider an array Arr which is to be sorted using Heap Sort.

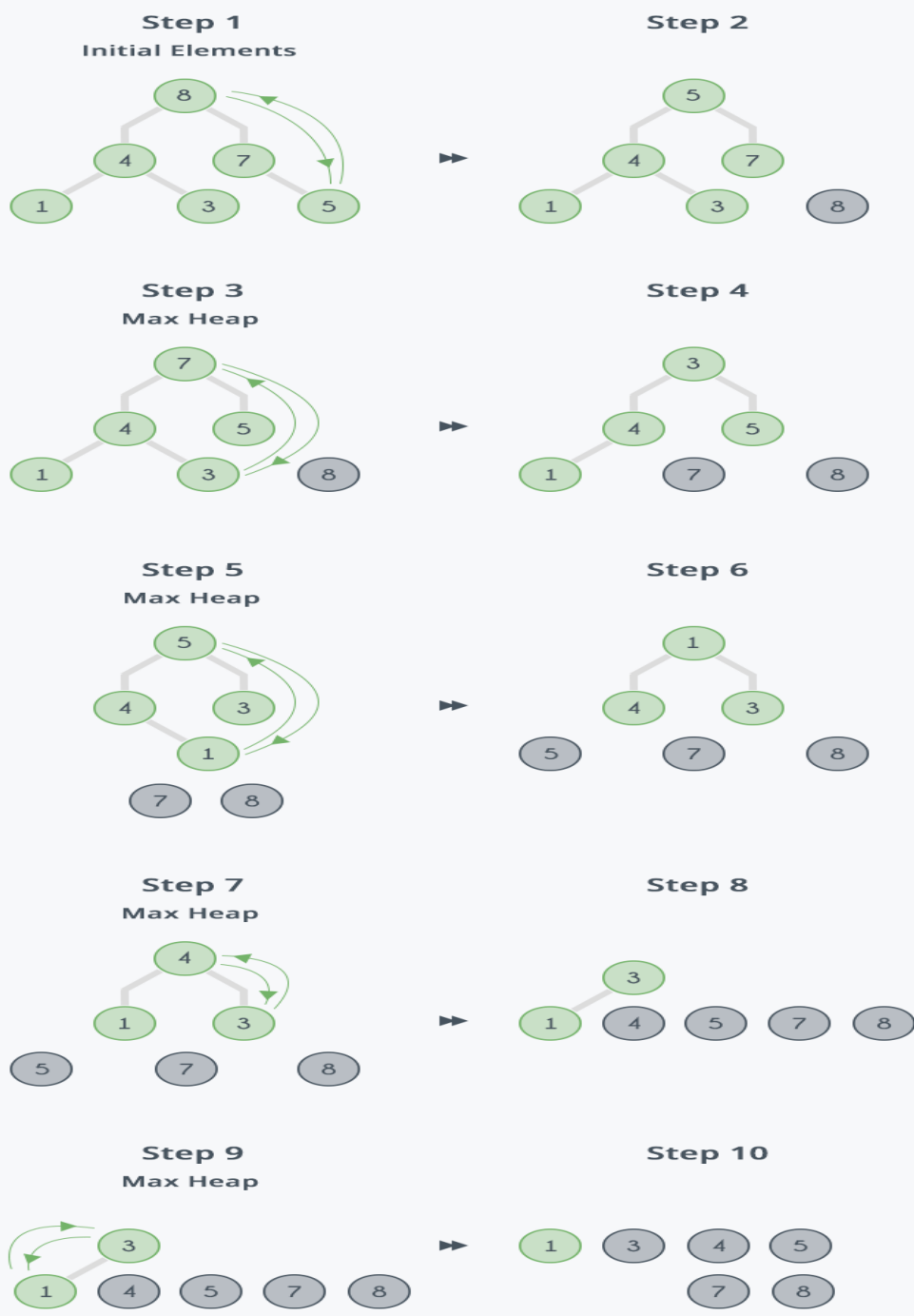
- Initially build a max heap of elements in Arr.
- The root element, that is Arr[1], will contain maximum element of Arr. After that, swap this element with the last element of Arr and heapify the max heap excluding the last element which is already in its correct position and then decrease the length of heap by one.
- Repeat the step 2, until all the elements are in their correct position

In the diagram below, initially there is an unsorted array Arr having 6 elements and then max-heap will be built.



After building maxheap, the elements in the array Arr will be:

Arr		8	4	7	1	3	5
	0	1	2	3	4	5	6



After all the steps, we will get a sorted array.

Arr		1	3	4	5	7	8
	0	1	2	3	4	5	6

## TIME COMPLEXITIES FOR HEAP SORT:

1. Best:  $\Omega(n \log n)$
2. Average:  $\theta(n \log n)$
3. Worst:  $O(n \log n)$