

Working of Heap sort Algorithm

Now, let's see the working of the Heapsort Algorithm.

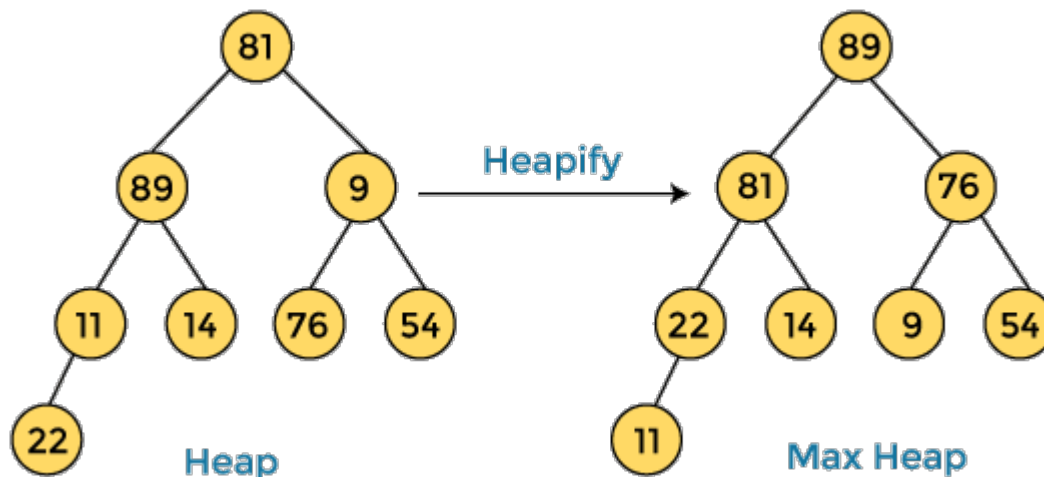
In heap sort, basically, there are two phases involved in the sorting of elements. By using the heap sort algorithm, they are as follows -

- The first step includes the creation of a heap by adjusting the elements of the array.
- After the creation of heap, now remove the root element of the heap repeatedly by shifting it to the end of the array, and then store the heap structure with the remaining elements.

Now let's see the working of heap sort in detail by using an example. To understand it more clearly, let's take an unsorted array and try to sort it using heap sort. It will make the explanation clearer and easier.

81	89	9	11	14	76	54	22
----	----	---	----	----	----	----	----

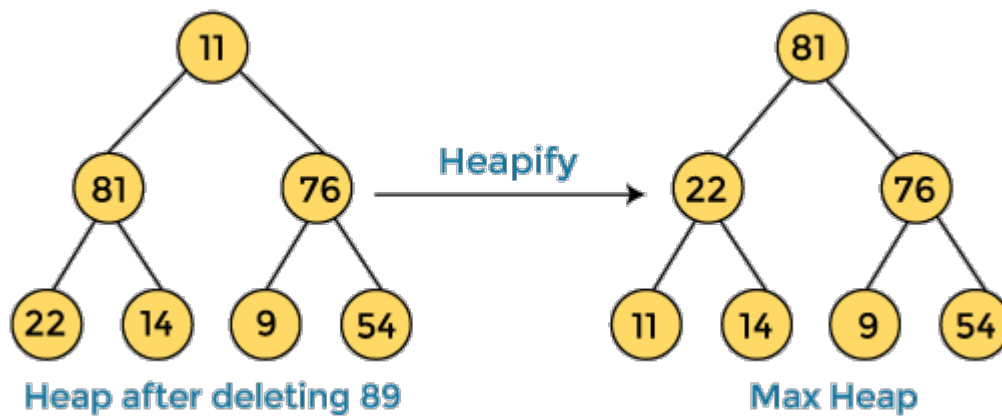
First, we have to construct a heap from the given array and convert it into max heap.



After converting the given heap into max heap, the array elements are -

89	81	76	22	14	9	54	11
----	----	----	----	----	---	----	----

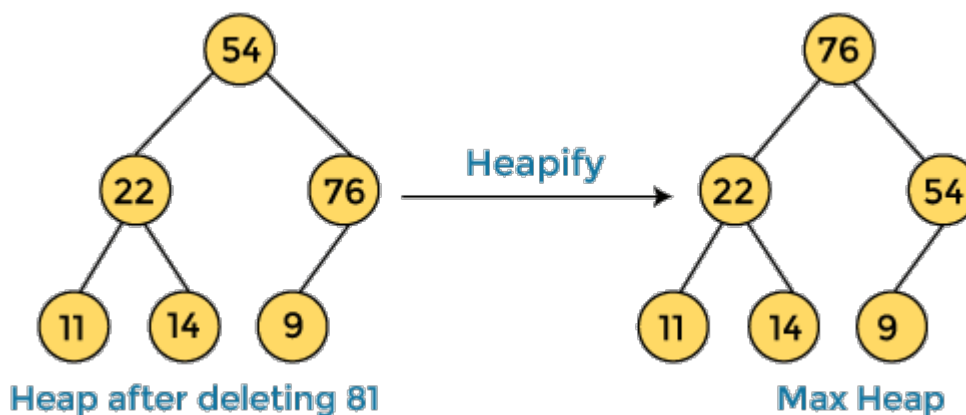
Next, we have to delete the root element (**89**) from the max heap. To delete this node, we have to swap it with the last node, i.e. (**11**). After deleting the root element, we again have to heapify it to convert it into max heap.



After swapping the array element **89** with **11**, and converting the heap into max-heap, the elements of array are -

81	22	76	11	14	9	54	89
----	----	----	----	----	---	----	----

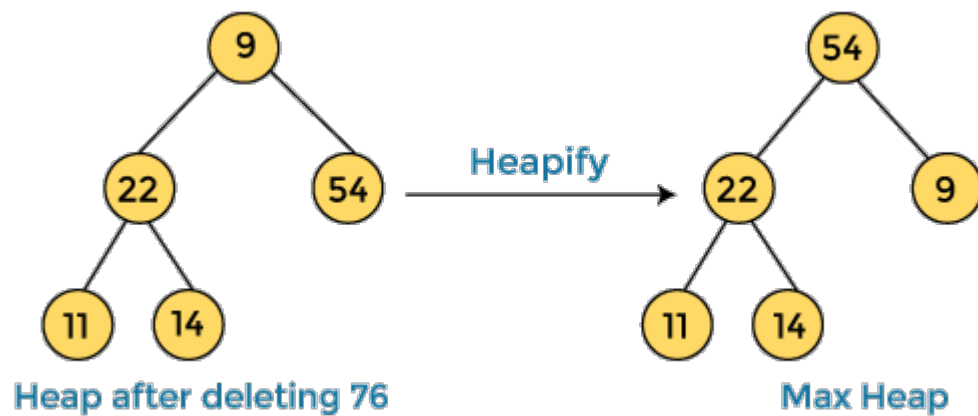
In the next step, again, we have to delete the root element (**81**) from the max heap. To delete this node, we have to swap it with the last node, i.e. (**54**). After deleting the root element, we again have to heapify it to convert it into max heap.



After swapping the array element **81** with **54** and converting the heap into max-heap, the elements of array are -

76	22	54	11	14	9	81	89
----	----	----	----	----	---	----	----

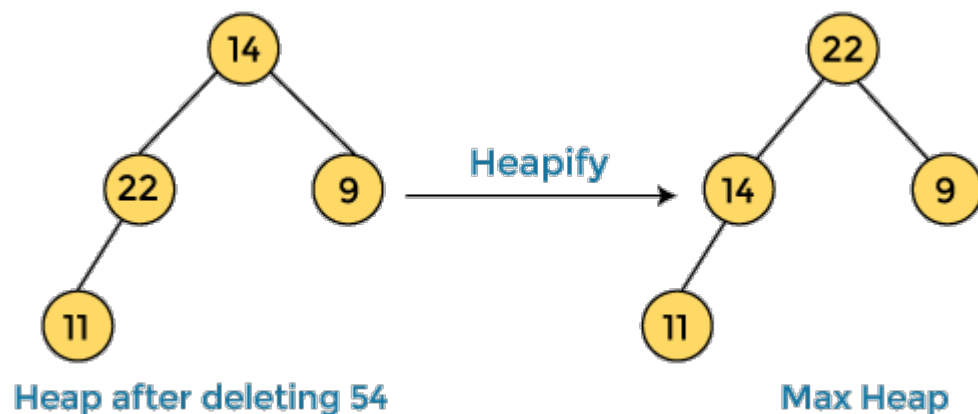
In the next step, we have to delete the root element (**76**) from the max heap again. To delete this node, we have to swap it with the last node, i.e. (**9**). After deleting the root element, we again have to heapify it to convert it into max heap.



After swapping the array element **76** with **9** and converting the heap into max-heap, the elements of array are -

54	22	9	11	14	76	81	89
----	----	---	----	----	----	----	----

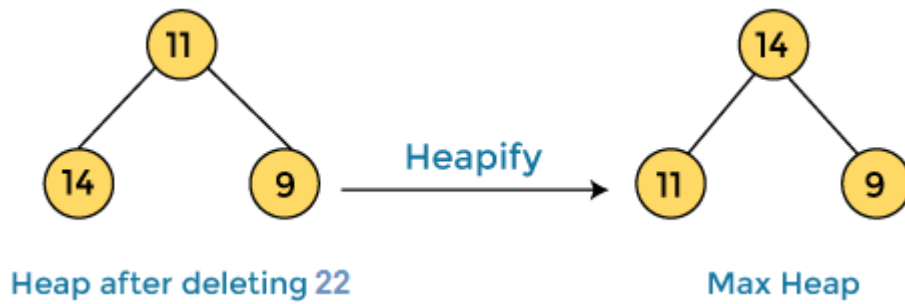
In the next step, again we have to delete the root element (**54**) from the max heap. To delete this node, we have to swap it with the last node, i.e. (**14**). After deleting the root element, we again have to heapify it to convert it into max heap.



After swapping the array element **54** with **14** and converting the heap into max-heap, the elements of array are -

22	14	9	11	54	76	81	89
----	----	---	----	----	----	----	----

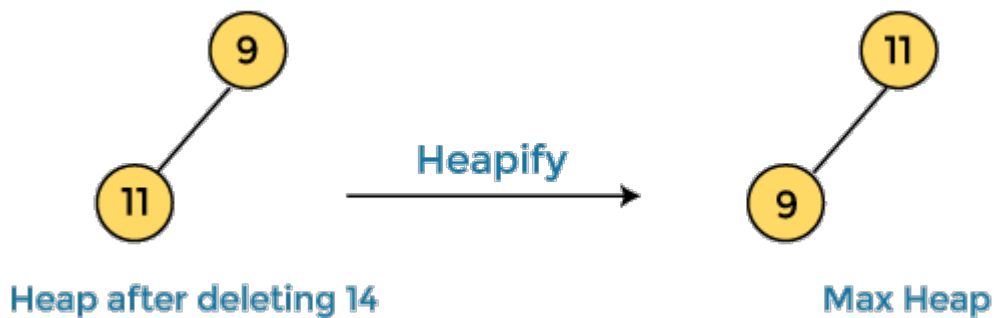
In the next step, again we have to delete the root element (**22**) from the max heap. To delete this node, we have to swap it with the last node, i.e. (**11**). After deleting the root element, we again have to heapify it to convert it into max heap.



After swapping the array element **22** with **11** and converting the heap into max-heap, the elements of array are -

14	11	9	22	54	76	81	89
----	----	---	----	----	----	----	----

In the next step, again we have to delete the root element (**14**) from the max heap. To delete this node, we have to swap it with the last node, i.e. (**9**). After deleting the root element, we again have to heapify it to convert it into max heap.



After swapping the array element **14** with **9** and converting the heap into max-heap, the elements of array are -

11	9	14	22	54	76	81	89
----	---	----	----	----	----	----	----

In the next step, again we have to delete the root element (**11**) from the max heap. To delete this node, we have to swap it with the last node, i.e. (**9**). After deleting the root element, we again have to heapify it to convert it into max heap.



After swapping the array element **11** with **9**, the elements of array are -

9	11	14	22	54	76	81	89
---	----	----	----	----	----	----	----

Now, heap has only one element left. After deleting it, heap will be empty.



After completion of sorting, the array elements are -

9	11	14	22	54	76	81	89
---	----	----	----	----	----	----	----

Now, the array is completely sorted.