

Heap Sort Algorithm

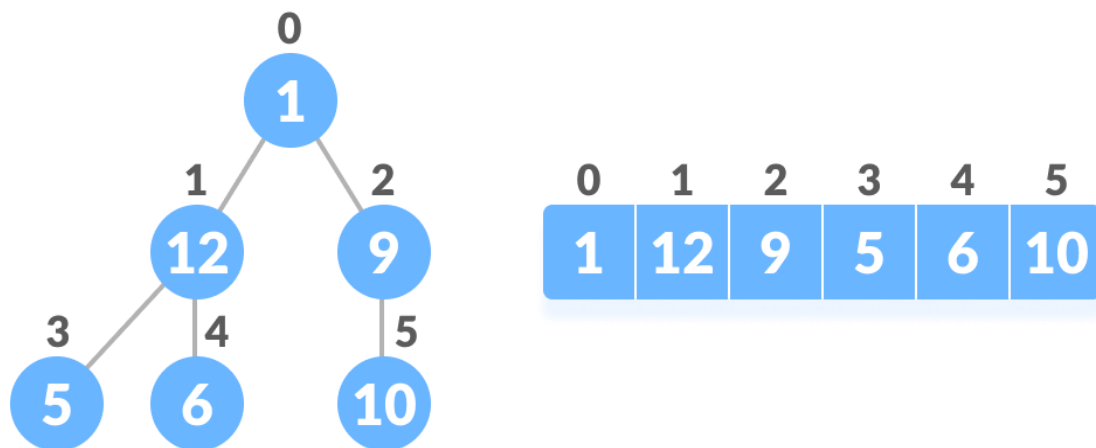
Heap Sort is a popular and efficient [sorting algorithm](#) in computer programming.

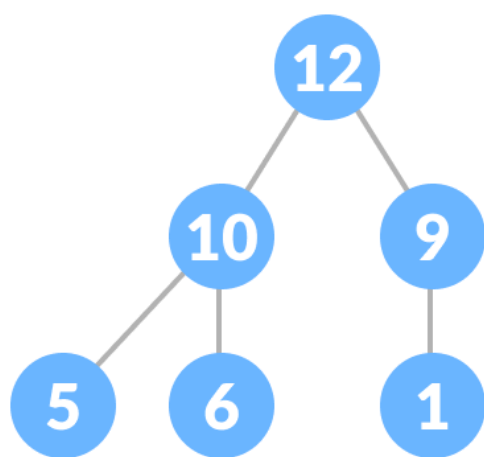
Learning how to write the heap sort algorithm requires knowledge of two types of data structures - arrays and trees.

The initial set of numbers that we want to sort is stored in an array e.g. [10, 3, 76, 34, 23, 32] and after sorting, we get a sorted array [3,10,23,32,34,76].

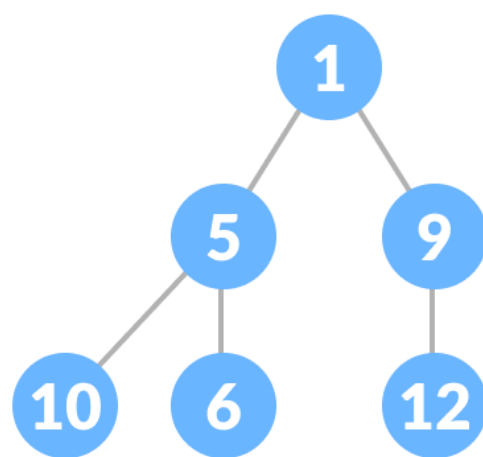
Heap sort works by visualizing the elements of the array as a special kind of complete binary tree called a heap.

Note: As a prerequisite, you must know about [a complete binary tree](#) and [heap data structure](#).



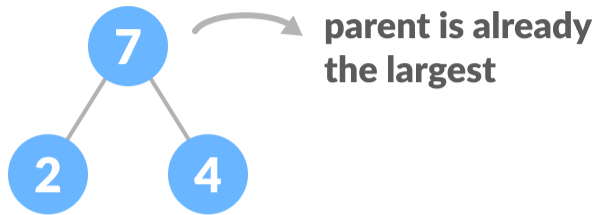


Max Heap

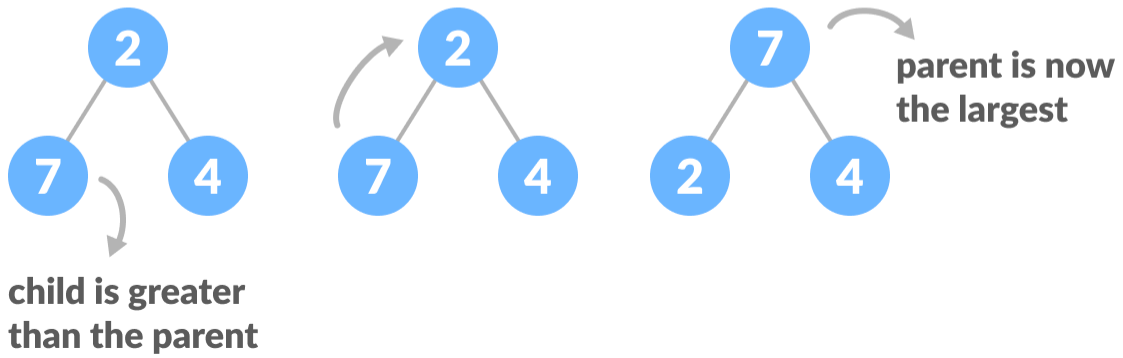


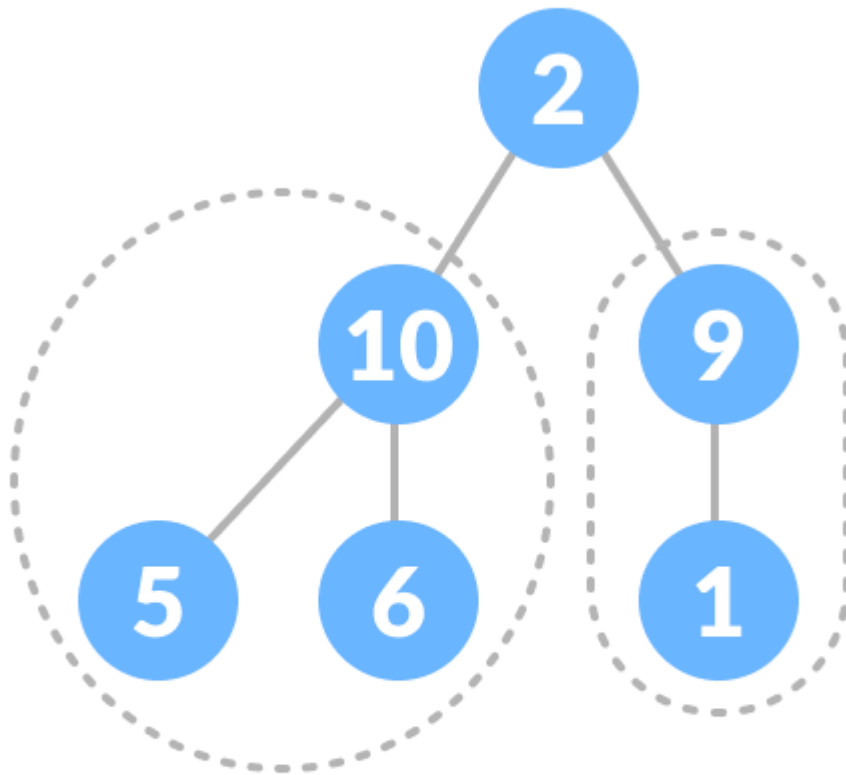
Min Heap

Scenario-1

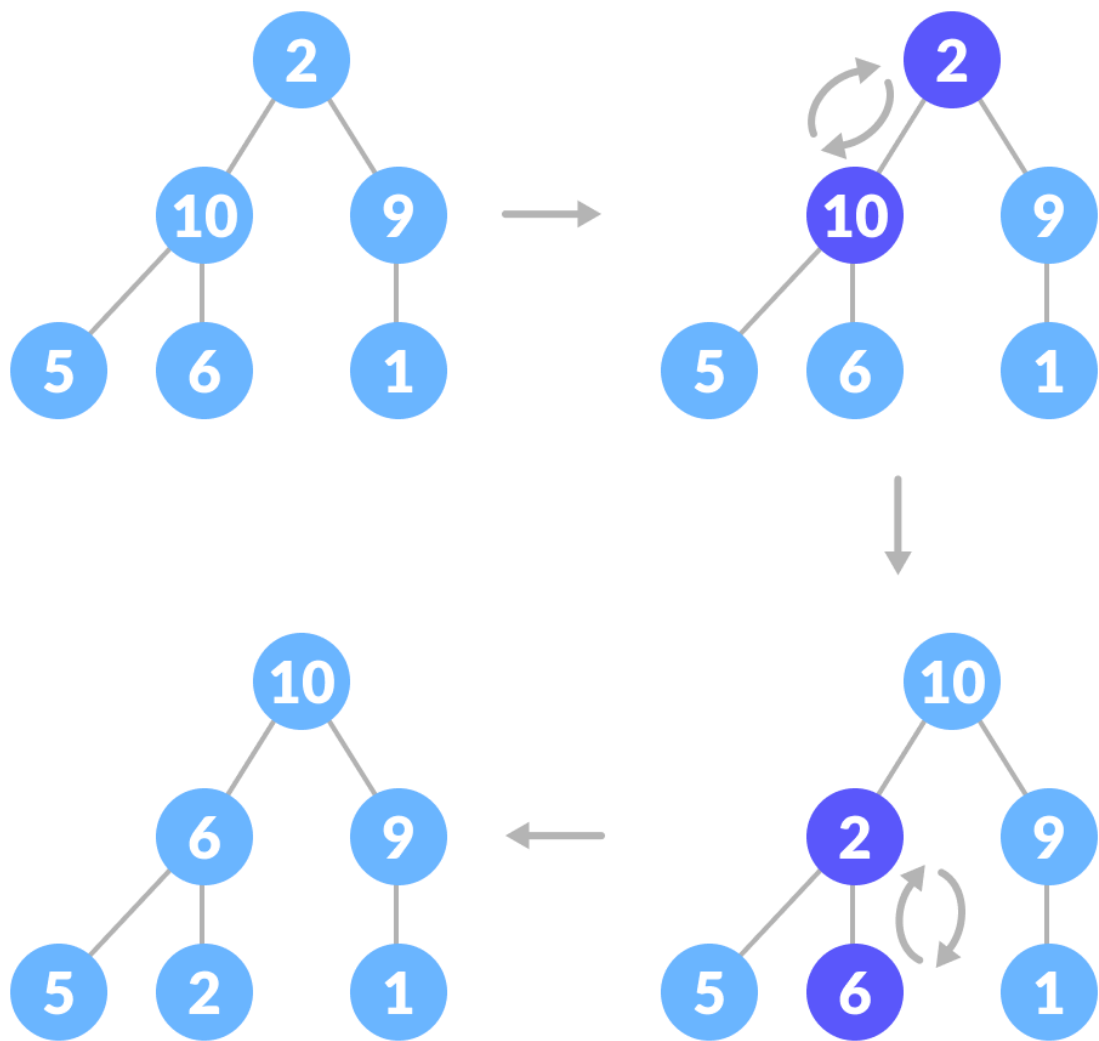


Scenario-2

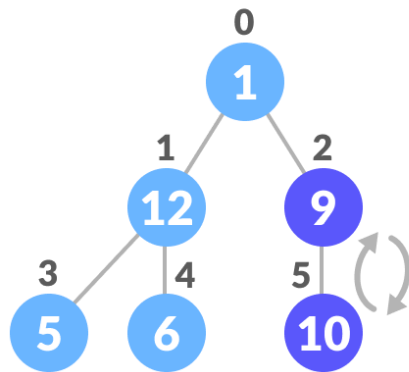




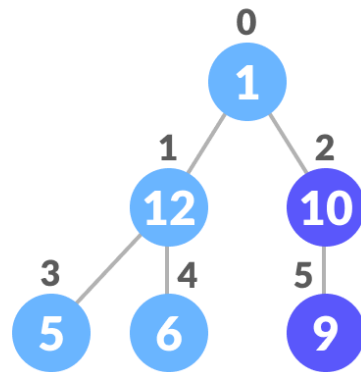
**both subtrees of the root
are already max-heaps**



$i = 2 \rightarrow \text{heapify}(\text{arr}, 6, 2)$

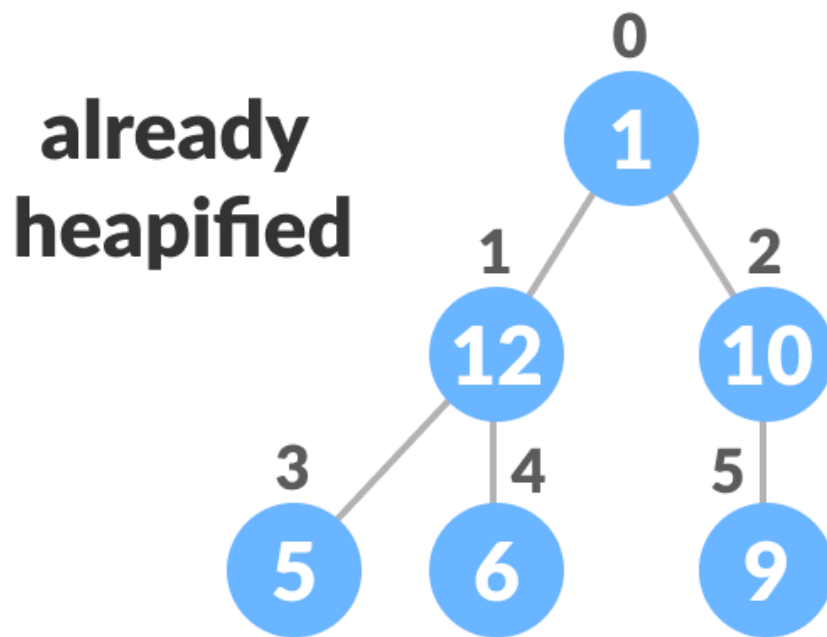


0	1	2	3	4	5
1	12	9	5	6	10

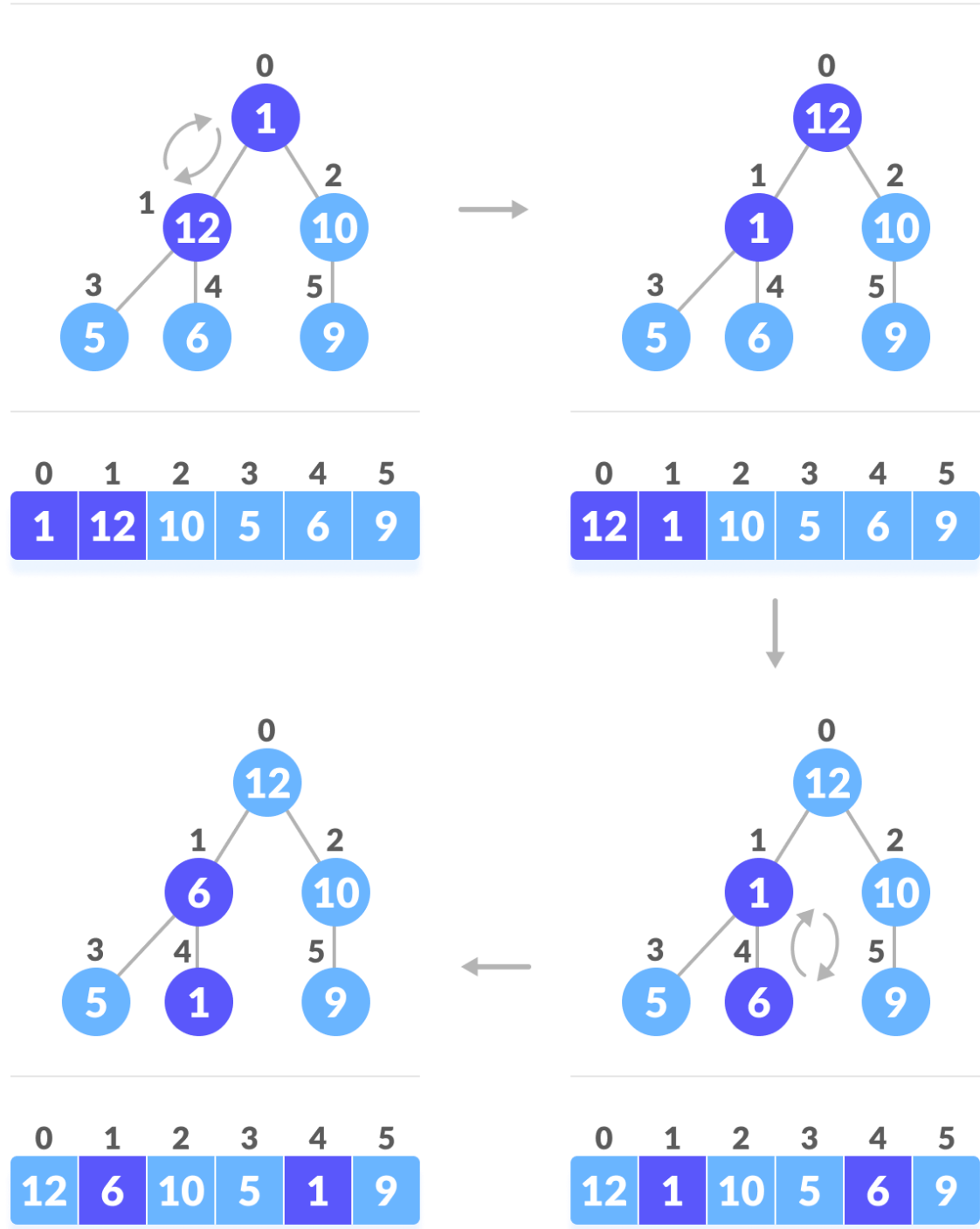


0	1	2	3	4	5
1	12	10	5	6	9

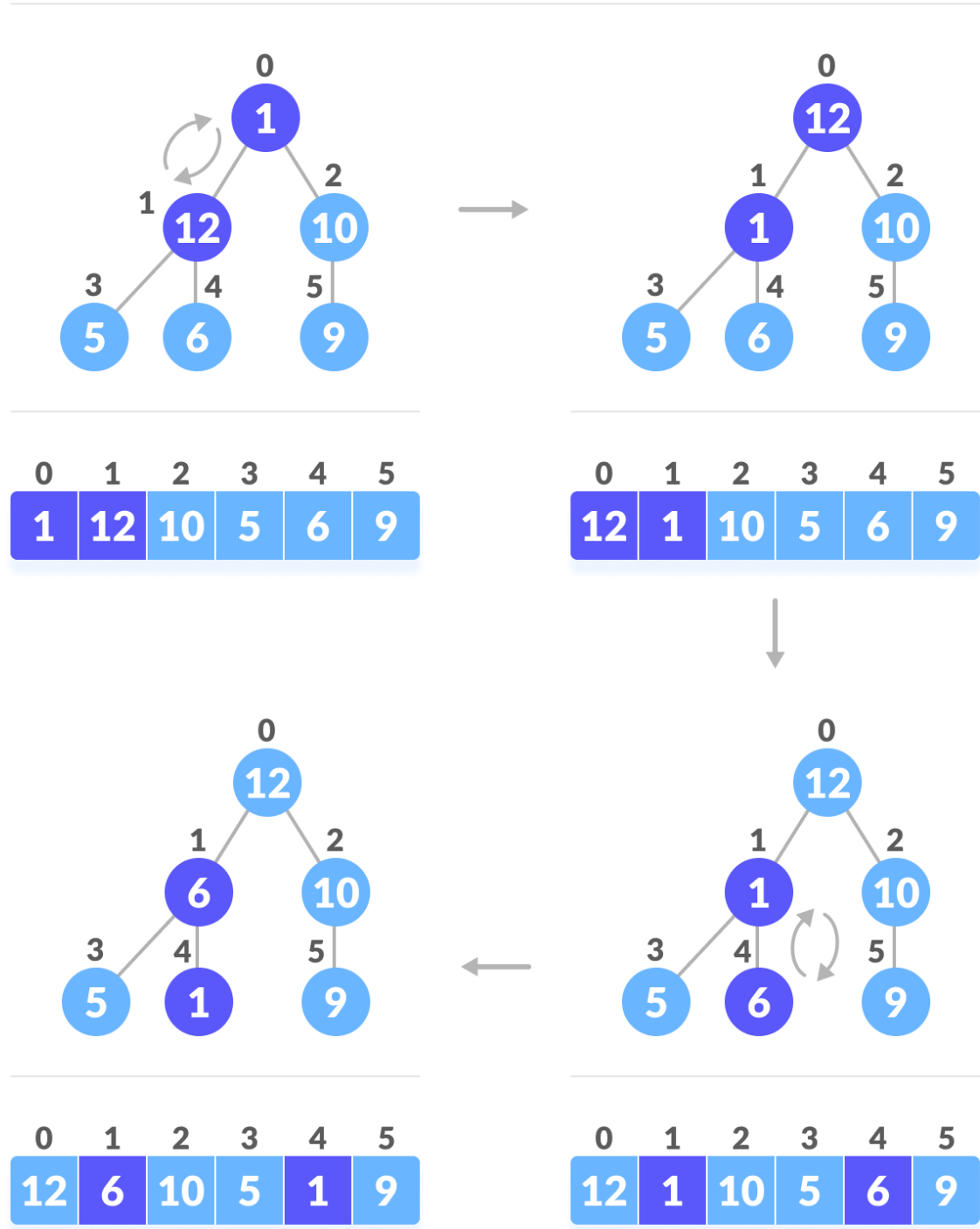
$i = 1$ \longrightarrow `heapify(arr, 6, 1)`



$i = 0 \rightarrow \text{heapify}(\text{arr}, 6, 0)$



$i = 0 \rightarrow \text{heapify}(\text{arr}, 6, 0)$



Working of Heap Sort

1. Since the tree satisfies Max-Heap property, then the largest item is stored at the root node.
2. **Swap:** Remove the root element and put at the end of the array (nth position) Put the last item of the tree (heap) at the vacant place.
3. **Remove:** Reduce the size of the heap by 1.
4. **Heapify:** Heapify the root element again so that we have the highest element at root.
5. The process is repeated until all the items of the list are sorted.

