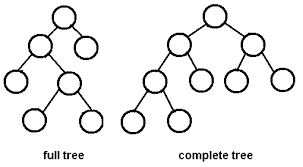
# Heap-Sort (Divide and Conquer)

1. **INTRODUCTION** :

a. ***Definition of a complete Binary Tree***:

A Tree is said to be complete binary tree if all it's levels have maximum number of nodes(two for each parent) except possibly the last level and all the nodes in the last level appear as per left as possible.

Example-



b***. Heap:***

A Heap is a Complete Binary Tree which is implemented using an array(Linear Data Structure).

* **Max Heap:** A heap is called max heap if all the nodes having a value greater than every children of the node.

This is used for Sorting in Ascending Order.

* **Min Heap:** a heap is called min heap if all the nodes having smaller value than every children of the node

1. Algoritm:

Algoritm for heap sort comprises of mainly Three parts .They are

1.take the unsorted **n** elements and Heapify them (in any max or min heap)

2.change first and last elements and adjust heap for remaining element leaving the last element.

3.Repeat step 2 until only 2 elements are there in the heap tree.

Algorithm Heap\_sort(array,n)

{

Make\_Heap(array,n);//Making Max heap

for (i:= n to 2 step-1)

{

T=array[1];

Array[1]=array[i];

Array[i]=T;

Adjust\_Heap(a,1,i-1);

}

}

Algorithm Make\_Heap(array,n)

{

for (i:=floor( n/2) to 1 step-1)

Adjust\_Heap(a,i,n);

}

Algorithm Adjust\_Heap(array,i,n)

{

j=2\*ai;//left child index

Item=array[i];//Parent's element

While(j<=n)

{

If((j<n)and(array[j]<array[j+1]))

j:=j+1;

If(item>array[j])

Break;

array[floor(j/2)]=array[j];

j=j\*2;//checking for child Heap

}

array[j/2]=item;

}

1. Time Complexity:

Time complexity of heapify is O(Logn). Time complexity of createAndBuildHeap() is O(n) and overall time complexity of Heap Sort is O(nLogn).

1. **Best Case:**

O(nlogn)

1. **Average Case:**

O(nlogn)

1. **Worst Case:**

O(nlogn)

1. Space Complexity:

O(n) - In this the space required is n spaces for array operations

It is not Stable When a new element comes it need to rearrange the entire Heap.

**Note : Applications of HeapSort**  
**1.** [Sort a nearly sorted (or K sorted) array](https://www.geeksforgeeks.org/nearly-sorted-algorithm/)2. [k largest(or smallest) elements in an array](https://www.geeksforgeeks.org/k-largestor-smallest-elements-in-an-array/)

As it has better time Complexity in all cases over quick and merge sort so it is more widely used