**Why heaps?**

Suppose we want to get min element from an array.But it will take O(n) time for unsorted time.

Now if we sort the array after inserting each element, then it can return min element in O(1) time. But sorting will take O(nlog(n)) time. Can we reduce this to O(logn) time?

Yes, using heap.

In Java, Heap is a special type of data structure where the root node or parent node is compared with its left and right children and arranged according to the order. Suppose, x is a root node and y is the child node, property ****key(x)<= key(y)**** will generate min-heap, and that relation is referred to as ****"Heap Property"****.

**Each node has at most two children, which are stored at positions 2\*i + 1 and 2\*i + 2 (assuming 0-based indexing), where i is the index of the current node in the array.**

Based on the order of the parent and child nodes, Heap can be classified in two forms, i.e., Min heap and Max heap. Let's understand both of them one by one and implement the code in Java.

## Min heap

Min heap is a special type of heap data structure that is a complete binary tree in itself . Min heap has the following properties:

1. Root node value is always smaller in comparison to the other nodes of the heap.
2. Each internal node has a key value that is always smaller or equal to its children.

## Max heap

Max heap is another special type of heap data structure that is also a complete binary tree in itself in Java. Max heap has the following properties:

1. Root node value is always greater in comparison to the other nodes of the heap.
2. Each internal node has a key value that is always greater or equal to its children.

**Heap implementation :** priority queue

