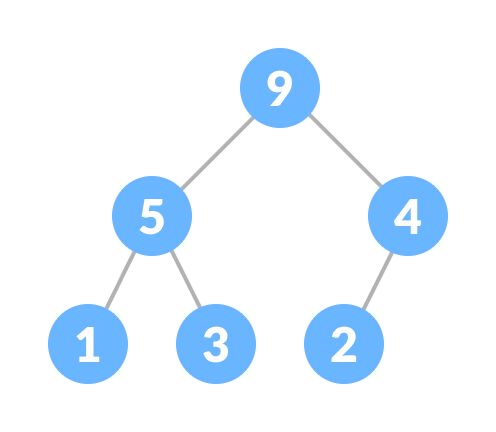
**Heap Data Structure:**

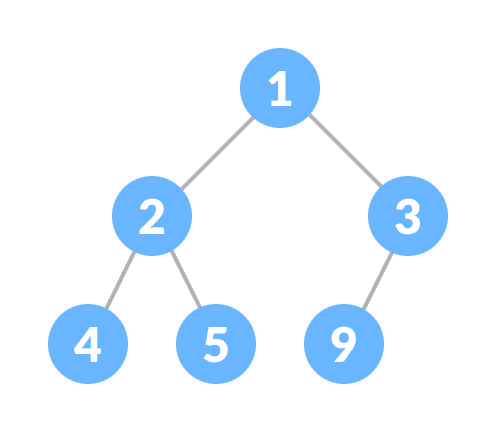
A Heap is a special Tree-based data structure in which the tree is a complete binary tree.

**Types:**

1. **Max-Heap**: In a Max-Heap the key present at the root node must be greatest among the keys present at all of its children. The same property must be recursively true for all sub-trees in that Binary Tree.

****

1. **Min-Heap**: In a Min-Heap the key present at the root node must be minimum among the keys present at all of its children. The same property must be recursively true for all sub-trees in that Binary Tree.



**Operation of heap:**

**Heapify**

Heapify is the process of creating a heap data structure from a binary tree. It is used to create a Min-Heap or a Max-Heap.

**Heap Data Structure Applications**

* Heap is used while implementing a priority queue.
* Dijkstra's Algorithm
* Heap Sort

**PRIORITY QUEUE:**

A priority queue is a **special type of queue** in which each element is associated with a **priority value**. And, elements are served on the basis of their priority. That is, higher priority elements are served first.

However, if elements with the same priority occur, they are served according to their order in the queue.

**Assigning Priority Value**

Generally, the value of the element itself is considered for assigning the priority. For example,

The element with the highest value is considered the highest priority element. However, in other cases, we can assume the element with the lowest value as the highest priority element.

Priority queue can be implemented using an array, a linked list, a heap data structure, or a binary search tree. Among these data structures, heap data structure provides an efficient implementation of priority queues.

A comparative analysis of different implementations of priority queue is given below.

|  |  |  |  |
| --- | --- | --- | --- |
| Operations | peek | insert | delete |
| Linked List | O(1) | O(n) | O(1) |
| Binary Heap | O(1) | O(log n) | O(log n) |
| Binary Search Tree | O(1) | O(log n) | O(log n) |

**Priority Queue Applications**

Some of the applications of a priority queue are:

* Dijkstra's algorithm
* for implementing stack
* for load balancing and interrupt handling in an operating system
* for data compression in Huffman code