Minimum and Maximum Value in Binary Search Tree .

Minimum element : Minimum element is nothing but **leftmost node** in binary search tree, so traverse left until you get leftmost element .

Maximum Element :

Maximum element is nothing but rightmost node in binary search tree, so traverse right until you get rightmost element.

Binary Heap Introduction :

Types :

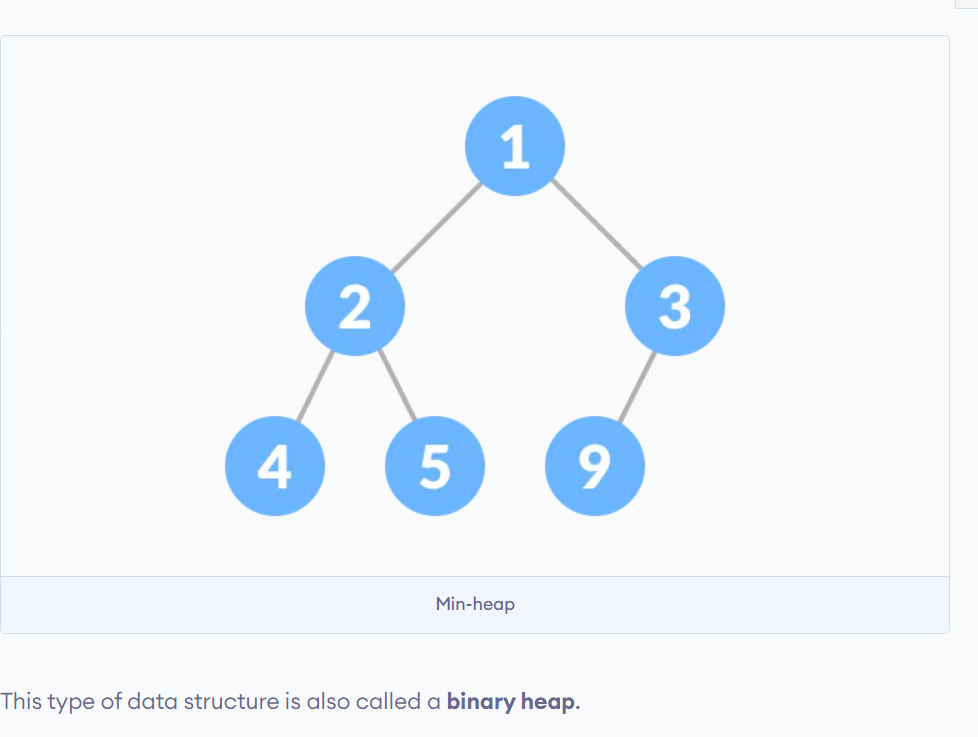
1. Binary Heap
2. Binomial Heap
3. Fibonacci Heap

* Binary Heap : The complete binary tree which satisfies the heap property .

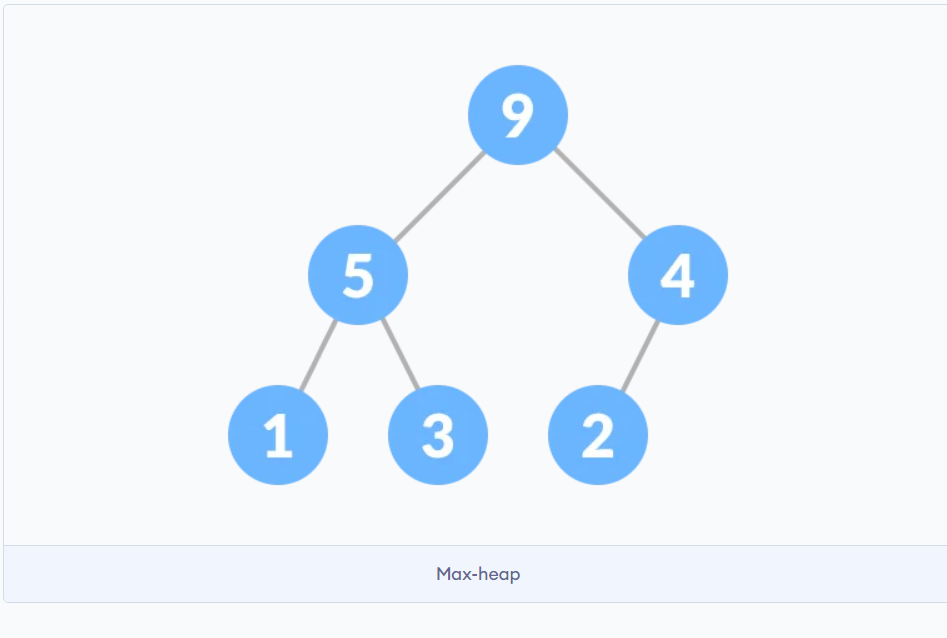
**Note: What is Heap Property ?**

**Ans : Heap Property is a property of a node in which key of every parent node need to be lesser than or eql to OR greater than or eql to the child nodes key .**

1. Min heap / min binary heap :



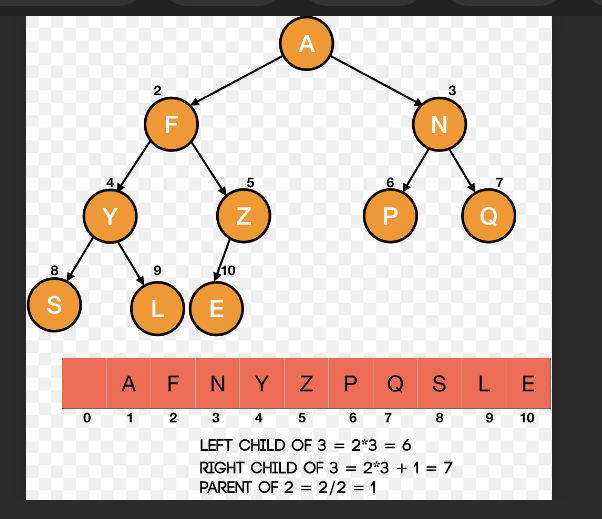
* 1. Complete binary tree,where the key of every parent node less than or equal to child node’s key .

1. Max heap / max binary heap :
   1. 
   2. Complete binary tree,where the key of every parent node greater than or equal to child node’s key.

Binary Heap Operations :

1. Heapify :
   1. It is a process to rearrange the elements of heap in order to maintain the heap property .
   2. 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.
      1. Types :
         1. Heapify\_up
         2. Heapify\_down
2. Insertion :
   1. Inserting a new node to binary heap by maintaining ist’s properties .
   2. Add the new node to first open spot available in the lower level .
   3. Heapify the new node.(check heapify property is violating or not )
3. Deletion :
   1. Removing the node from binary heap by maintaining its properties.
   2. Swap the node you want to delete with the last node.
   3. Delete last node .
   4. Heapify the last node which is now placed in the deleted node position.

List Representation of Binary Tree :



Formula for calculating the Parent node of Given node :

Parent = (i-1) // 2

Where i = index of gien node

* Find left child of given node :

Left child = ( 2 \* i )

* Find right child of given node :

Right child = ( 2 \* i ) + 1

* Heap implementation In Python

Note:

1. Floor Division :
   1. Floor division works same as normal division .( eg. 10//5 = 2 )
   2. If the resulting number is in decimal form we got to round that off to a nearest integer which is less than the resulting integer .
   3. Eg . 9//2 = 4.5 = 4
2. Importants Formulas :
   1. Max. no. of nodes = (2^(height of tree)+1) -1
   2. Height of Tree = [log (max. Num. of node)]
   3. Leaf node of Complete binary tree :

[ n/2 ] + 1 to n

Where n = number of nodes