

### **DAA LAB – 3**

**AIM:** To create a binary tree, display in order walk of tree, find maximum and minimum depth of binary tree, compute the lowest common ancestor of binary search tree and to implement heap sort by varying the number of input elements and compute the time and memory.

**A Binary Search Tree (BST)** is a tree in which all the nodes follow the below-mentioned properties:

- The left sub tree of a node contains only nodes with keys lesser than the node's key.
- The right sub tree of a node contains only nodes with keys greater than the node's key.
- The left and right sub tree each must be a binary search tree.

**Heap Sort:** Heaps can be used in sorting an array. In max-heaps, maximum element will always be at the root. Heap Sort uses this property of heap to sort the array. Heap sort is a comparison-based sorting technique based on Binary Heap data structure.

#### **Observation:**

N	10	100	500	1000	10000
Time	1.00	1.0025	1.0074	1.011	1.087
Memory	8.46484375	8.48046875	8.48828125	8.51171875	8.8125

**Conclusion:** Successfully compared runtime and memory utilization for different inputs of heap sort, Implemented in order walk for binary search tree and calculated maximum, minimum depth and lowest common ancestor of nodes of BST.