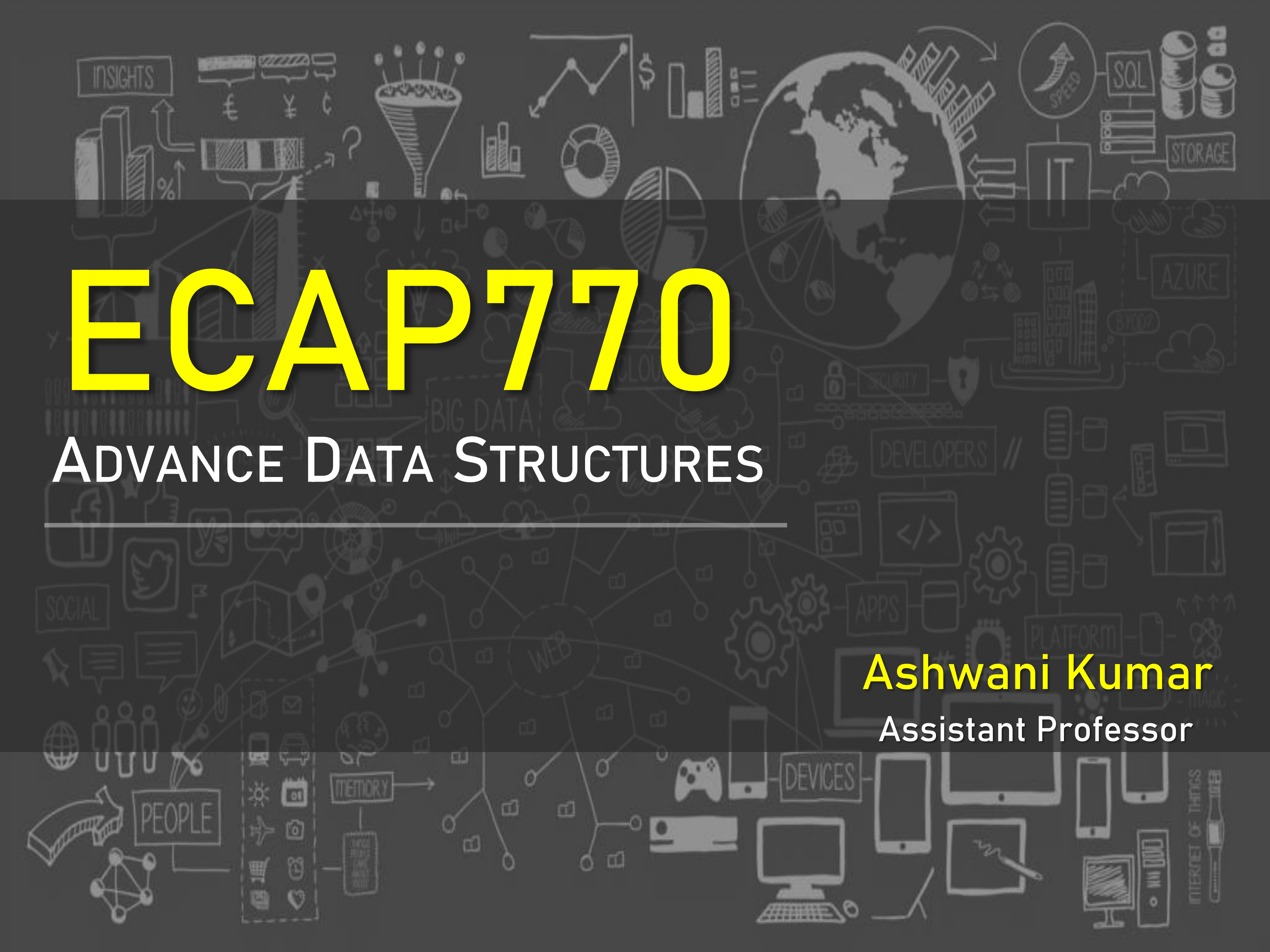


ECAP770

Advance Data Structures

Ashwani Kumar

Assistant Professor

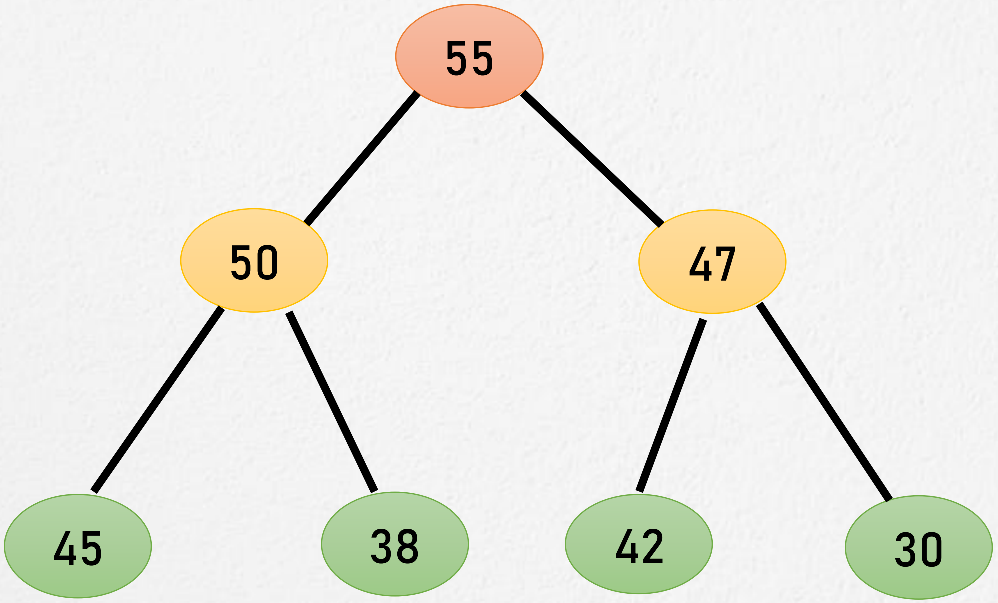
Learning Outcomes

After this lecture, you will be able to

- Understand tree data structure
- Understand binary search tree

Tree Data Structure

A tree is a nonlinear hierarchical data structure that consists of nodes connected by edges.

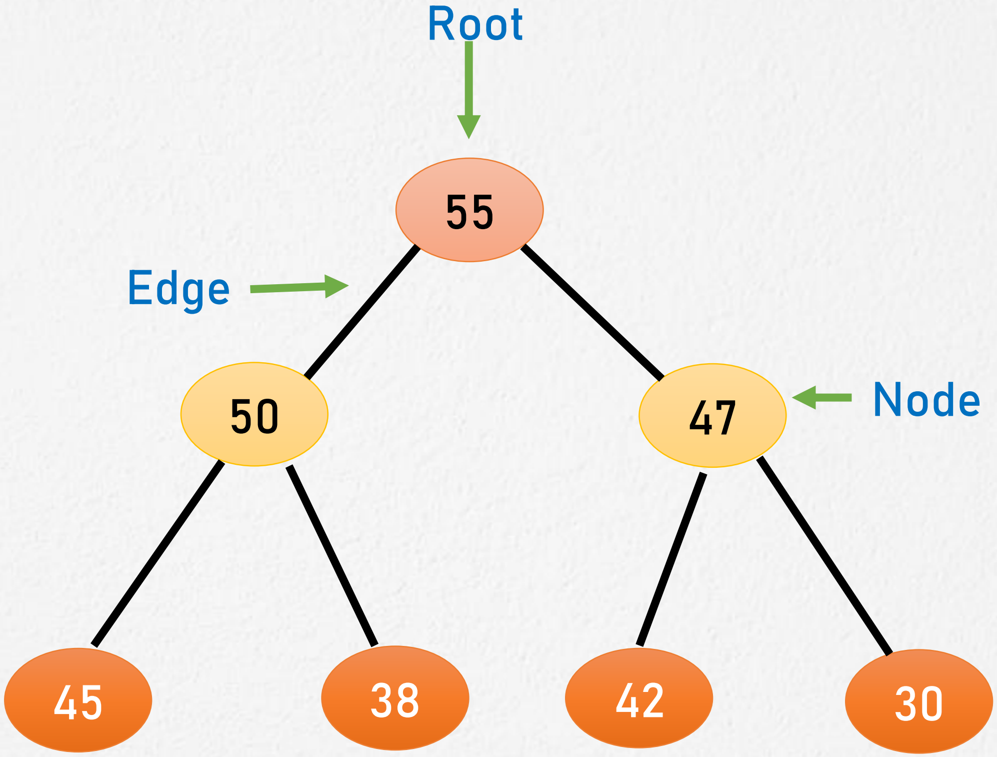


Tree Data Structure

- Other data structures such as arrays, linked list, stack, and queue are linear data structures that store data sequentially.
- In order to perform any operation in a linear data structure, the time complexity increases with the increase in the data size.
- Different tree data structures allow quicker and easier access to the data as it is a non-linear data structure.

Tree Terminologies

- Node
- Edge
- Root



Types of Trees

- General Tree
- Binary Tree
- Binary Search Tree
- AVL Tree

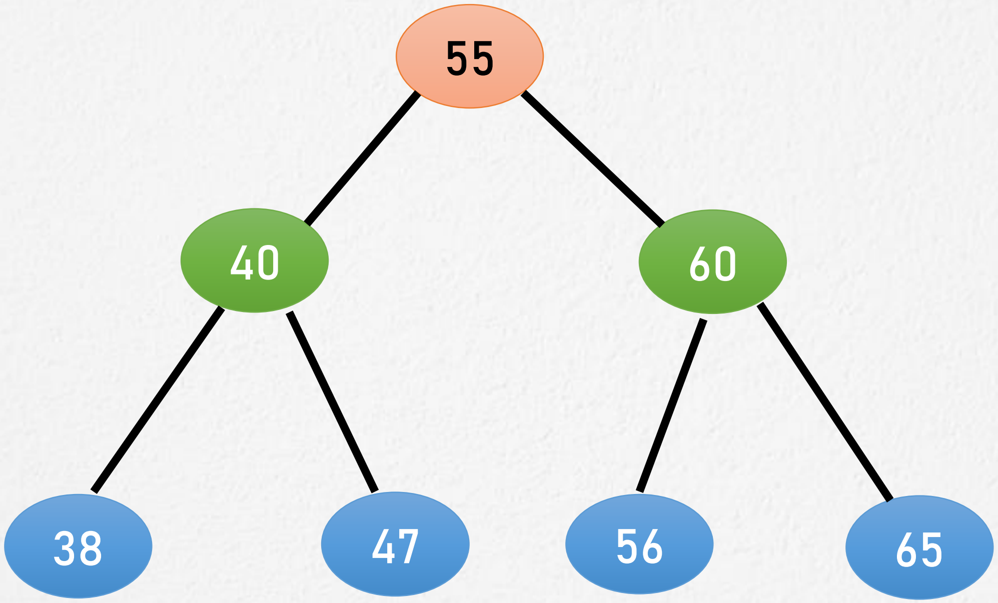
Binary search tree

- Binary search tree is a non-linear data structure in which one node is connected to n number of nodes. It is a node-based data structure.
- It is also called an ordered or sorted binary tree.

Binary search tree

- In a binary search tree, the value of all the nodes in the left sub-tree is less than the value of the root.
- Similarly, value of all the nodes in the right sub-tree is greater than or equal to the value of the root.

Binary search tree



Types of Binary Trees

- Complete binary tree: All the levels in the trees are full of last level's possible exceptions. Similarly, all the nodes are full, directing the far left.
- Full binary tree: All the nodes have 2 child nodes except the leaf.
- Balanced or Perfect binary tree: In the tree, all the nodes have two children. Besides, there is the same level of each sub node.

Binary tree creation

- List of elements: 55 40 60 45 38 56 65
- Insert 55 into the tree as the root of the tree.
- Read the next element, if it is lesser than the root node element, insert it as the root of the left sub-tree.
- Otherwise, insert it as the root of the right of the right sub-tree.

Binary tree creation

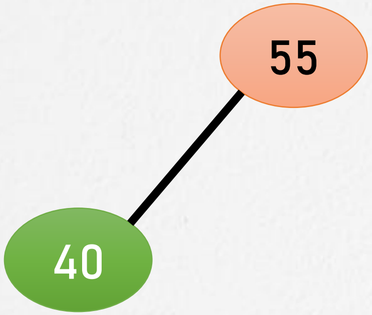
List of elements:

55 40 60 45 38 56 65

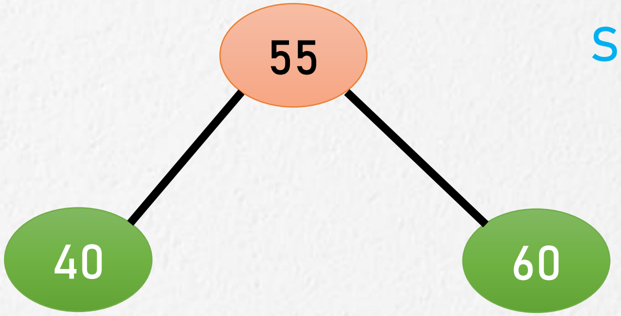
Step 1

Binary tree creation

Step 2



55 40 60 45 38 56 65

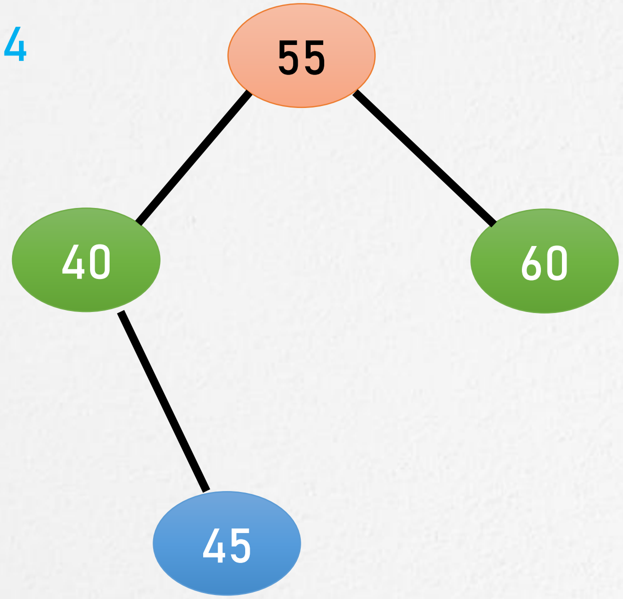


Step 3

55 40 60 45 38 56 65

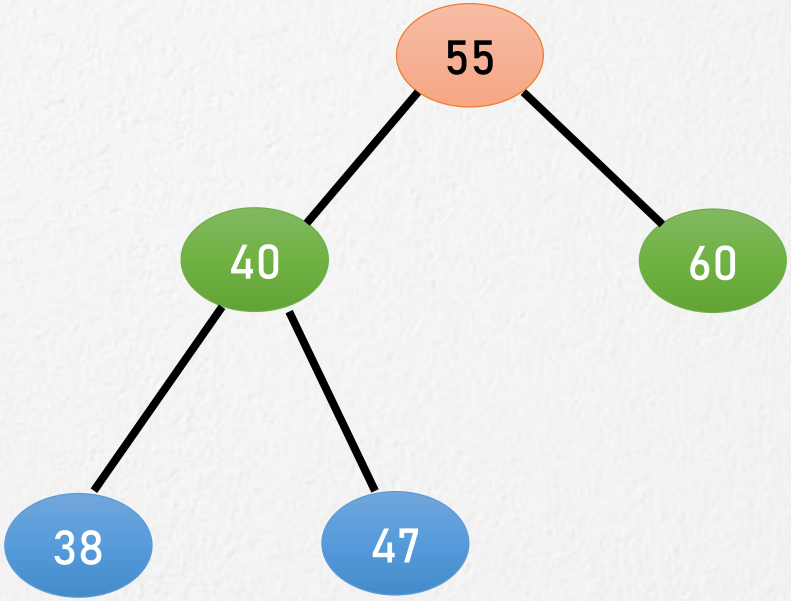
Binary tree creation

Step 4



55 40 60 45 38 56 65

Step 5

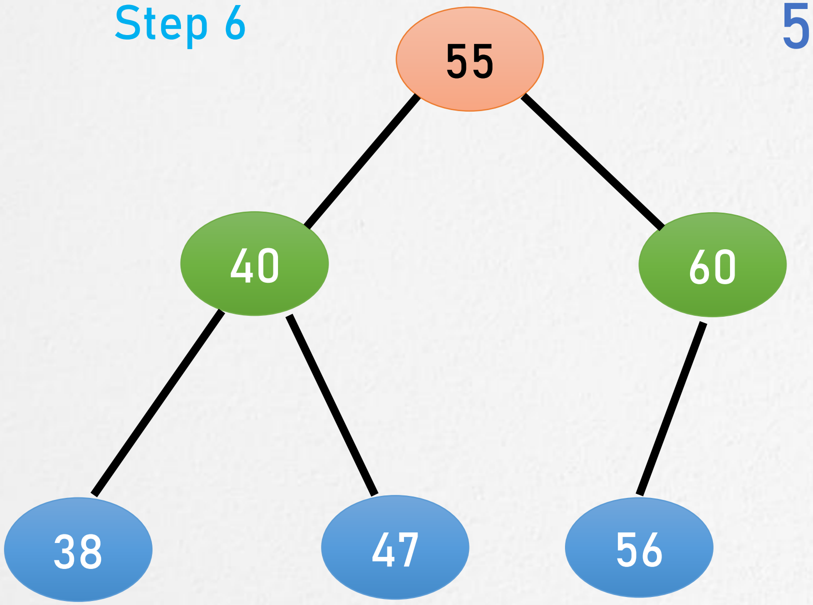


55 40 60 45 38 56 65

Binary tree creation

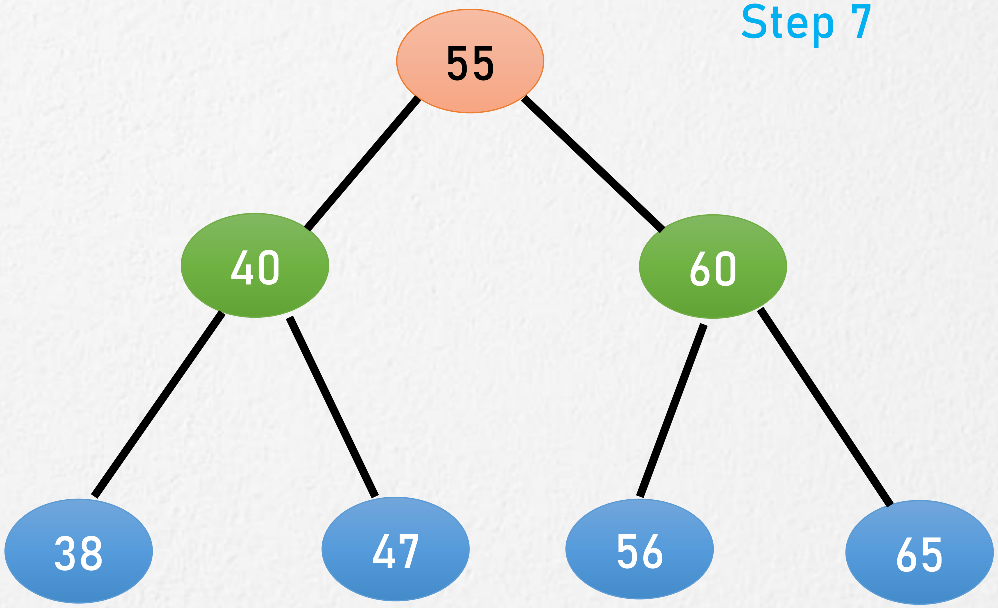
Step 6

55 40 60 45 38 56 65



Step 7

55 40 60 45 38 56 65



Binary Tree Operations

Search

Insertion

Deletion

Advantages of binary search tree

- Searching become efficient in a binary search tree. it is easy to identify which sub-tree contains the desired element.
- As compare to arrays and linked lists binary search tree is considered as efficient data structure.
- In searching process, it removes half sub-tree at every step. Searching for an element in a binary search tree takes o(log2n) time.
- It also speed up the insertion and deletion operations as compare to that in array and linked list.

Binary Search Tree Applications

- In multilevel indexing in the database.
- For dynamic sorting.
- For managing virtual memory areas in Unix kernel.
- It is used to implement various searching algorithms.

That’s all for now…