

Binary Search Tree (BST)

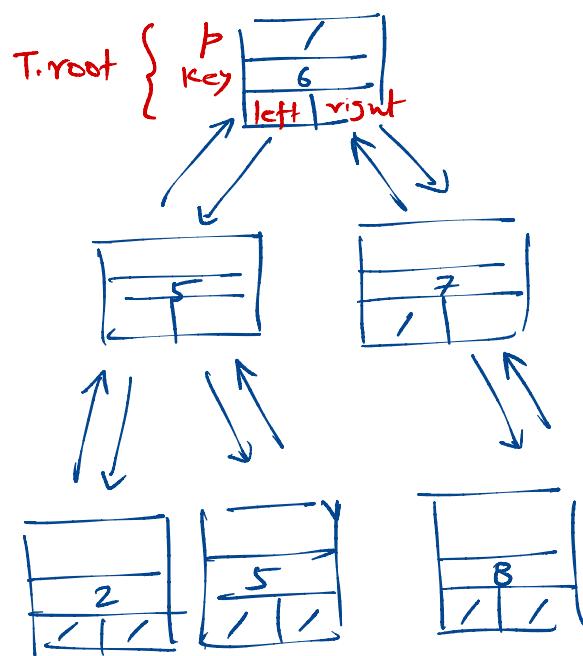
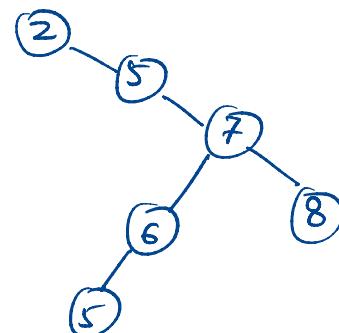
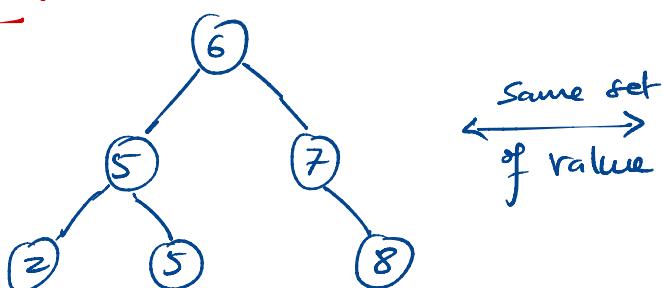
- A search tree is a Dynamic Set that supports operations like: search, insert, delete, minimum, maximum, predecessor, successor, etc.
- Basic operations in a BST take time proportional to the height of the tree.
 - for a complete binary tree $\sim \Theta(\lg n)$
- A BST is organized in a binary tree
- Can be represented with a linked data structure (\neq linked list)
 - fields: key, left, right, $\text{P} \rightarrow$ points to the parent
 - value
 - points to left child
 - points to right child
 - If a child or parent is missing, the attribute contains a NIL value
- Tree has a root attribute that points to the root of the tree, or NIL if the tree is empty.
 - The root $T.\text{root}$ is the only node whose Parent is NIL

- Keys in a BST satisfy the Binary Search Tree property

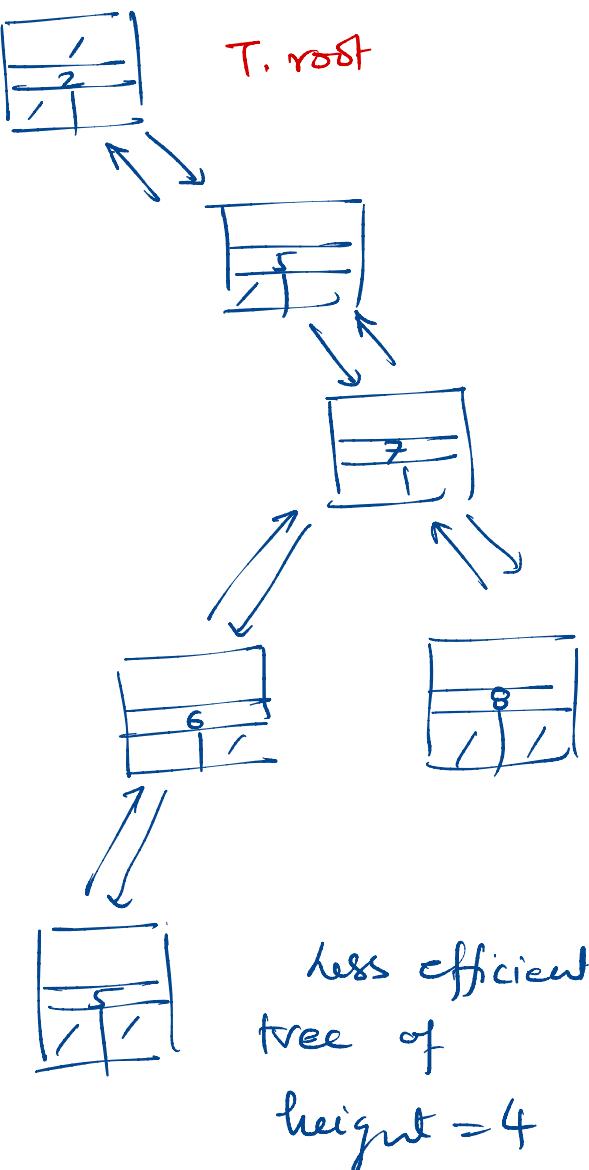
"Let x be a node in a BST. If y is a node in the left subtree of x , then $y.key < x.key$.

If y is a node in the right subtree of x , then $y.key \geq x.key$.

Example :



height = 2



BST traversal

Due to properties of BST

- Possible to print all keys in a BST in sorted order by performing an inorder tree walk

Print the root key of a subtree after printing values in the left subtree and before printing values in the right subtree.

Tree traversals

- Depth first search (DFS): Explore the tree in depth before going to the next sibling.
 - Pre order walk
 - In order walk
 - Post order walk

Pre order walk:

walk order: TLR

1. visit root first
2. recursively traverse the left subtree
3. recursively traverse the right subtree

In order walk:

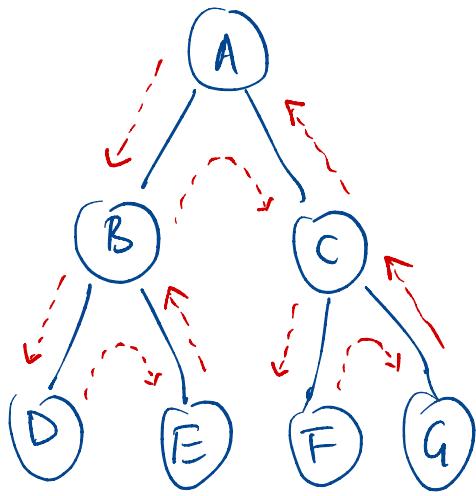
walk order: LTR

1. recursively traverse the left subtree
2. Visit the root
3. recursively traverse the right subtree

Post order walk:

walk order: LRT

1. recursively visit the left subtree
2. recursively visit the right subtree
3. visit the root



Pre order walk (TLR) :

A B D E C F G

In order walk (LTR) :

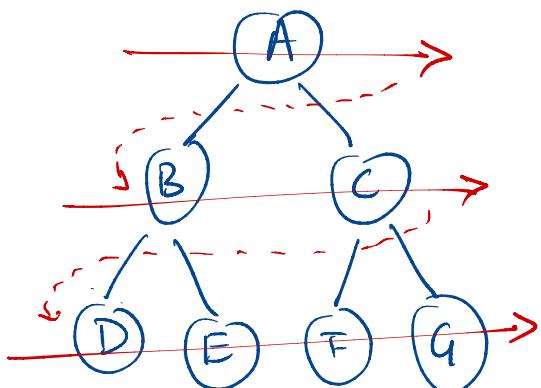
D B E A F C G

Post order walk (LRT) :

D E B F G A C A

} DFS walks

Breadth First Search (BFS) : The tree is broadened as much as possible before going to the next level
 (level-order search)



BFS walk : A B C D E F G

In order - Tree - walk (x)

if $x \neq \text{NIL}$

Inorder-tree-walk ($x.\text{left}$)

Print $x.\text{key}$

Inorder-tree-walk ($x.\text{right}$)
