Date.....
Page.....

DSA LAB-10

Theory

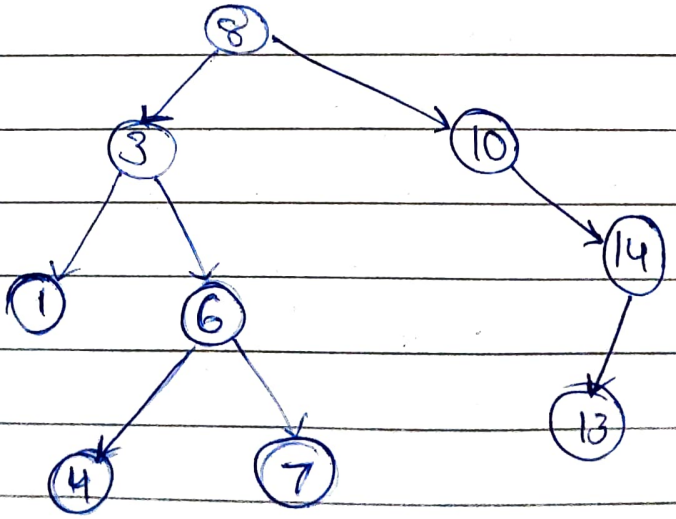
1) A. Binary search tree is a node-base binary tree data structure which has following properties.

→ The left subtrees of a node contains only nodes with keys lesser than the node's key

→ The right subtree of a node contains only nodes with keys greater than the node's key

→ The left and right subtree each must be a binary search tree.

Ex:



Date.....
Page.....

Binary tree has each internal node x, stores an element such that the element stored in the left subtree of x are less than or equal to x and elements stored in right subtree of x are greater than or equal to x. This is called binary-search-tree property.

2)A. Members of structure of tree are :

i) Elements are called 'Trees'.
ii) Lines connecting nodes are called 'Branches'.
iii) Nodes without children are called 'Leaf Nodes'.
iv) Nodes which have same parent node are called 'Sibling Nodes'.
v) Nodes which has no superior node or parent is called 'root'.

The size of tree is defined by total number of nodes in trees.

3) A. The rules to construct binary search tree is:

→ The left subtrees of a node contains only nodes with keys lesser than the node's key

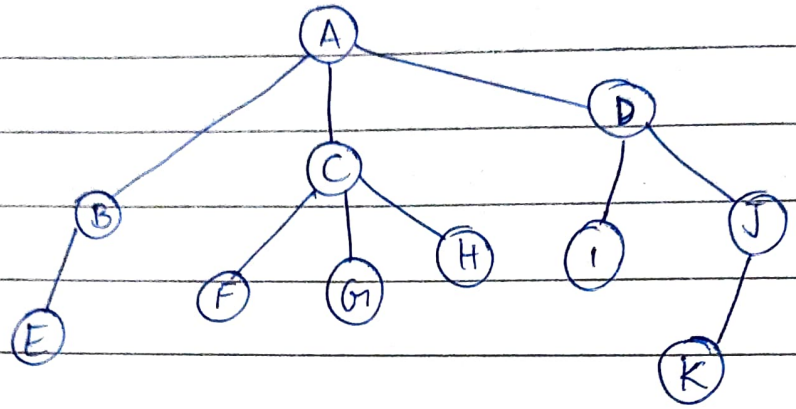
→ The right subtree of a node contains only nodes with keys greater than the node's key

→ The left and right subtree each must be a binary search tree.

→ The target node hat at most 2 child nodes.

Date
Page

4)A. Let us take a general tree example and convert into binary tree.



→ As per rules of binary search tree, the root node of general tree A is the root node of the binary tree.

→ Now the leftmost child node of the root node in the general tree is B and it is the leftmost child node of the binary tree.

→ Now as B has E as its leftmost child node, so it is leftmost child node in the binary tree whereas it has C as its rightmost sibling node so it isits rightmost child node inthe binary tree.

→ Now chas F as its leftmost childnode and D as its rightmost sibling node, So they are its left and right child node in

binary tree respectively

→ Now D has I as its leftmost child node which it left child node in the binary tree but doesn't have any rightmost sibiling node, so doesn't have any right child in the binary tree.

→ ~~Now for I as its leftmost child~~

→ Now for I, J is its rightmost sibling node so it is the right child node in binary tree.

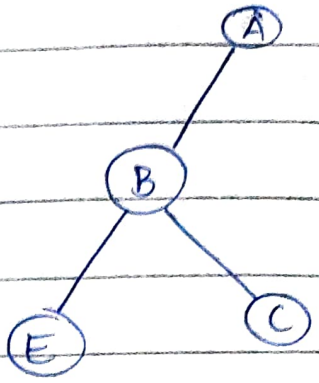
→ Similarly for J, K as its leftmost child node and thus it its left child node in binary tree.

→ Now for C, F is its leftmost child node, which has G as its rightmost sibling node, which has H as its right sibling node and thus they form their left, right and right child node respectively.

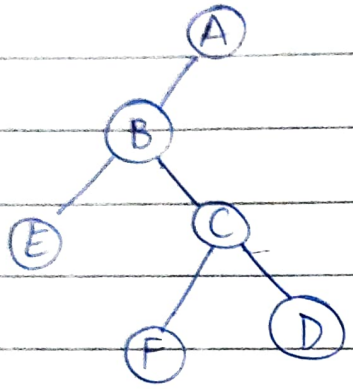
Date.....
Page.....

Step 1

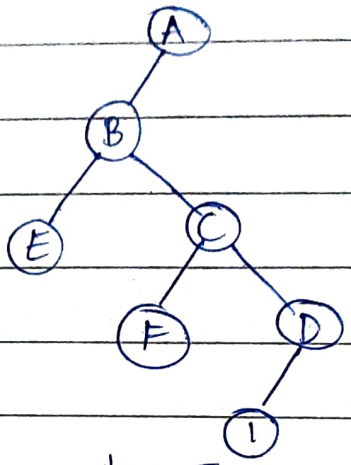
Step 2



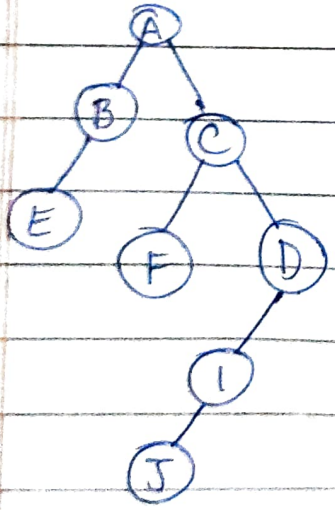
step 3



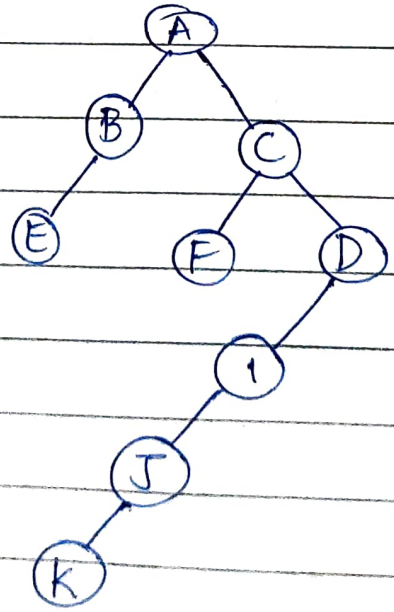
Step 4



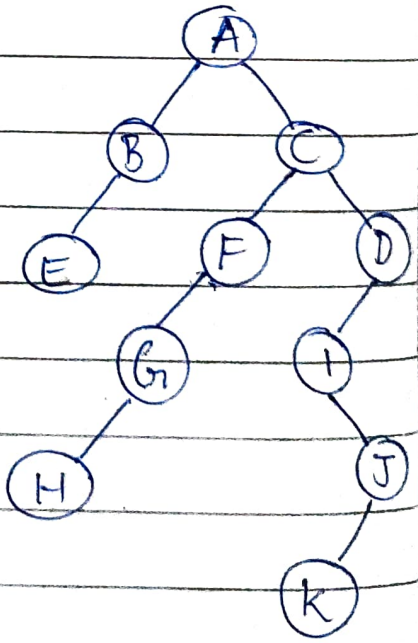
step 5



Step 6



Step 7



Step 8

5)A. Inorder traversal of a Binary tree can either be done using recursion or with the use of a auxiliary stack. The idea of threaded binary trees is to make inorder traversal faster and do it without stack and without recursion. A binary tree is made threaded by making all right child pointers that would normally be NULL point to the inorder successor of the node (if it exists.)

There are two types of threaded binary trees.

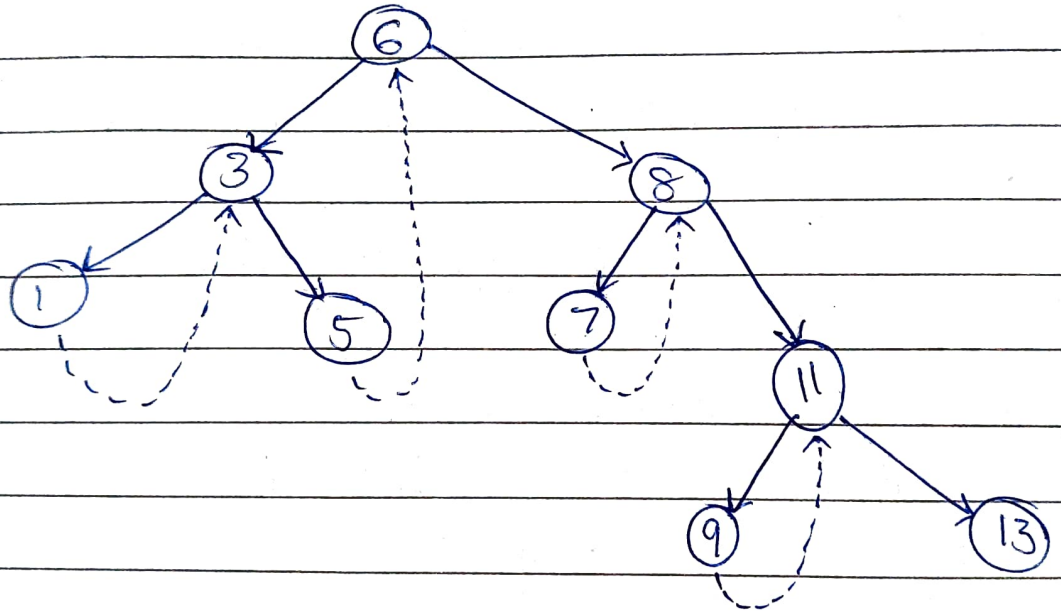
Single Threaded: where a NULL right pointers is made to point to the inorder successor (if successor exists)

Double Threaded: where both left and right NULL pointers are made to point to inorder predecessor and inorder successor respectively. The predecessor threads are useful for reverse inorder traversal and postorder traversal.

Date.....
Page.....

The threads are also useful for fast accessing ancestors of a node.

Following diagram shows an example Single Threaded Binary Tree. The dotted lines represent threads.



Date.....
Page.....

6)A. The idea of threaded binary trees is to make inorder traversal faster and do it without stacks and without recursion.

Threaded trees makes the traversal a little faster, because we are guaranteed O(1) time to access next node. In case of regular binary tree, it would take (o log n) time, because we have to climb up and down.