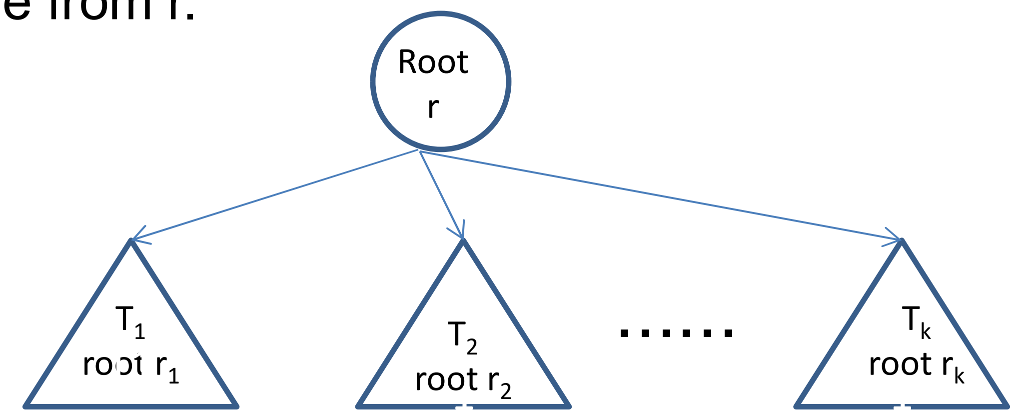
Lecture – 04 - Tree

- A tree T is a collection of n nodes.
  - If the collection is empty, then the tree is called a null tree.
  - One of the nodes, r is specially designated as root of the tree
  - The remaining nodes are partitioned in to k subsets, T_1, T_2, ..., T_k
  - Each subset represents a tree with r_1, r_2, ..., r_k as roots.
  - Each of these roots r_1, r_2, ..., r_k is connected through a directed edge from r.
-



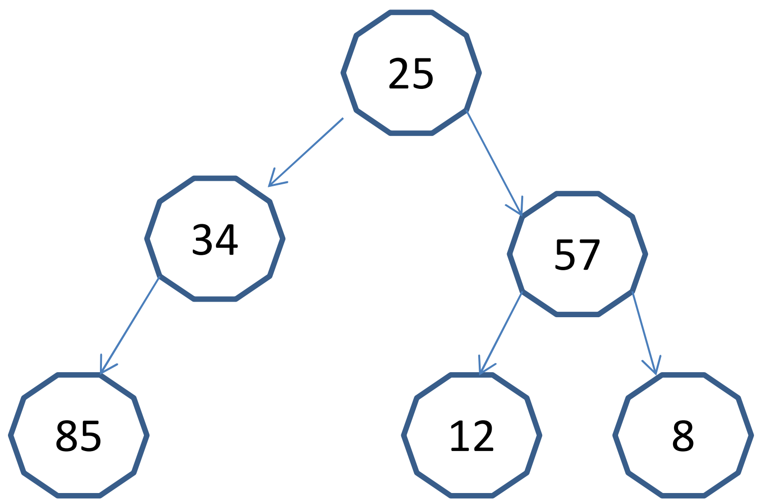
Binary Tree

- A node in a tree is connected to any number of nodes. These nodes are called child nodes and the node connecting them is called parent.
- A tree can be represented through a structure called First-child Next-sibling.
- This kind of a general tree has very limited applications
- A tree such that every nodes has at most two child nodes is called Binary tree.
- Any node in a binary tree can have
  - No children
  - Only left child
  - Only right child
  - Both children

Binary Tree

- A node in a tree can be represented as

- A binary tree with elements 25, 34, 67, 85, 12, 8 can be represented as



- In this tree, we can traverse in inorder, preorder or postorder.

Traversing in a binary tree

- Inoder
  - Traverse the left subtree
  - Traverse the root
  - Traverse the right subtree
- Preorder
  - Traverse the root
  - Traverse the left subtree
  - Traverse the right subtree
- Postorder
  - Traverse the root
  - Traverse the right subtree
  - Traverse the left subtree

Binary tree – level of nodes

- Root of a tree is considered as a node at level 0.
- Its immediate children are at level 1. Children of these nodes are at level 2 and so on.
- If a tree has n nodes,
  - what can be the maximum level of any node in the tree?
- Height of a tree
  - Height of a tree with one node is 1
  - Height of a tree with zero nodes is -1
  - Height of a tree is

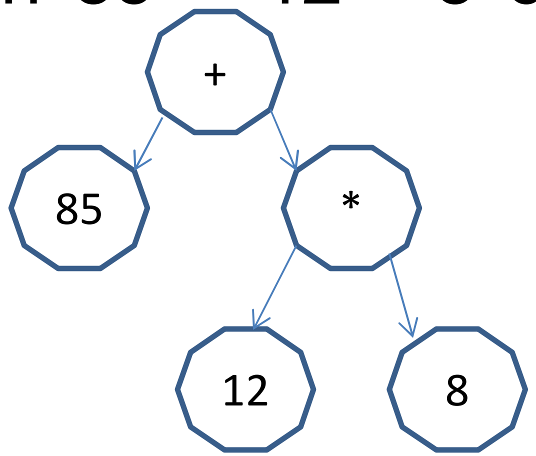
    1 + max(height(T → left), height(T → right))

Complete binary tree

- Nodes in a binary can be numbered as follows:
  - Root is numbered 1.
  - If any node is numbered as k, its left child is numbered as 2k and its right child is numbered as 2k + 1.
  - This is called node numbering.
- If the tree with n nodes has a node corresponding to each of the node numbers from 1 to n, then the tree is called complete binary tree.
- A complete binary tree can be implemented as an array.

Binary Tree

- A binary tree can also be used to represent a binary expression.
- An expression 85 + 12 * 8 can be represented as



Inorder: 85 + 12 * 8

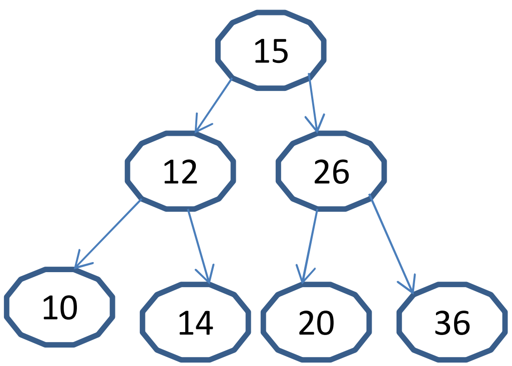
Postorder: 85 12 8 * +

Preorder: + 85 * 12 8

Inorder expression can be ambiguous. Preorder or postorder expressions are unambiguous

Binary Search Tree

- A binary search tree is a binary tree such that
  - ~~Data at every node is greater than that at its left child and less than that at its right child.~~
  - Data at every node is greater than that at every node in its left subtree
  - Data at every node is less than that at every node in its right subtree.



What happens if 14 is changed as 24?
Or 20 is changed as 10?

- Search, insert and delete operations can be performed in a binary search tree.
- In the above tree insert 18, search for 20, delete 14, delete 26

Binary Search Tree

- Insert 18:
  - Compare 18 with root. It is more. Move to right subtree
  - Compare 18 with 26. It is less. Move to left subtree.
  - Compare 18 with 20. It is less. Move to left subtree.
  - The left link in the node with data 20 is NULL. It is null tree.
  - Create a new node p, with data as 18, left and right links as null
  - Make p as left link of node 20.
- Search 20
  - Same as above.
  - Either the element is found or reach a null tree. In such case element is not present in the tree.

Binary Search Tree

- Delete 14, Delete 26
  - Search for 14 / Search for 26.
  - It can be
    - » a leaf node.
    - » Node with only left child
    - » Node with only right node
    - » Node with both children
  - In the first case, the node can be deleted.
  - In the second and third case, the node can be replaced with the left child or right child respectively
  - In the fourth case, the data at the node can be swapped with the data at the left most node (say node x) in the right subtree and node x can be deleted.

Binary Search Tree

- A sequence of numbers can be inserted in to an initially empty binary search tree.
- What can be the height of resultant BST?
- In the worst case, it can be n-1. How? Example?
- In the best case, it can be log n. How? Example?
- What is the number of comparisons for inserting an element in to a BST?
  - It can be maximum n – 1 and minimum is log n.
- Consider inserting the elements 23, 12, 45, 62, 98, 28?
- Reorder these elements and insert.
- The worst case time complexity for inserting or searching for an element is O(n).
- Can this be improved?

Height balanced tree

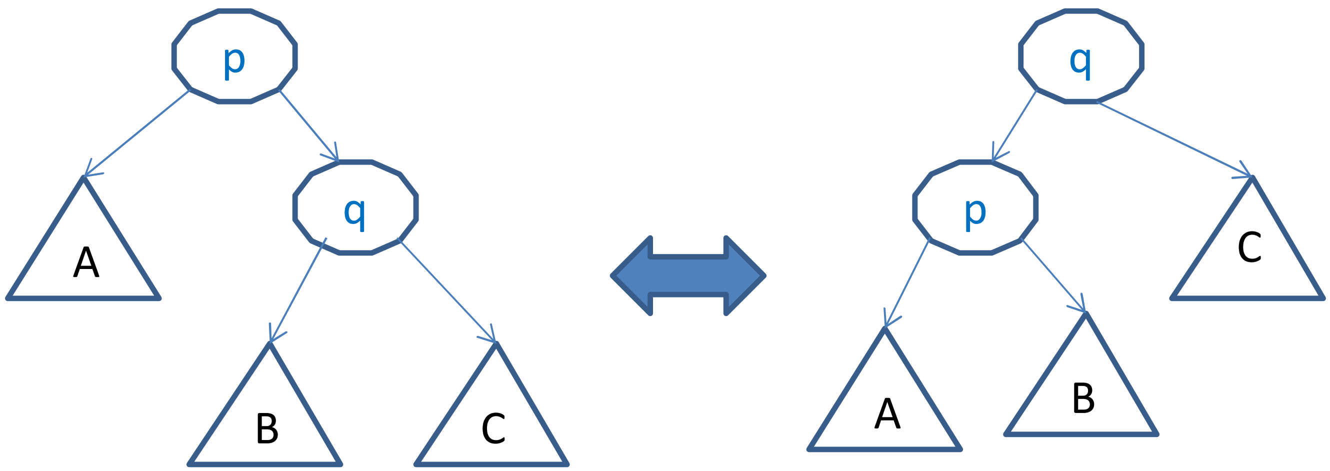
- If a sequence of numbers in increasing order are inserted in to an initially empty binary search tree, the resultant binary search tree can be with a height of n – 1
- If the elements can be inserted in some specific order, the maximum height of the resultant BST can be log n.
- Consider inserting 14, 8, 25, 4, 12, 36, 18 in to an initially null BST
- In the elements are inserted in the order 4, 8, 12, 14, 18, 25, and 36, what is the resultant BST?
- Can we adjust the height of a BST after each insertion, so that the heights of left subtree and right subtree of each node are balanced.
- The resultant search tree is called AVL - tree

AVL tree

- AVL (Adelson – Velskii – Landis) tree is a binary search tree with height balancing factor –
- At every node in the tree, the absolute difference between the heights of its left subtree and right subtree is at most 1.
- This involves rotation of the tree whenever the balancing factor is violated at any node.
- These rotations can be
  - single rotation, involving a node, its right child and the subtrees of these nodes.
  - single rotation, involving a node, its left child and the subtrees of these nodes.
  - Double rotation involving a node, its left child, its right child and the subtrees of these nodes.
  - Double rotation involving a node, its right child, its right child and the subtrees of these nodes.

Rotations in AVL tree

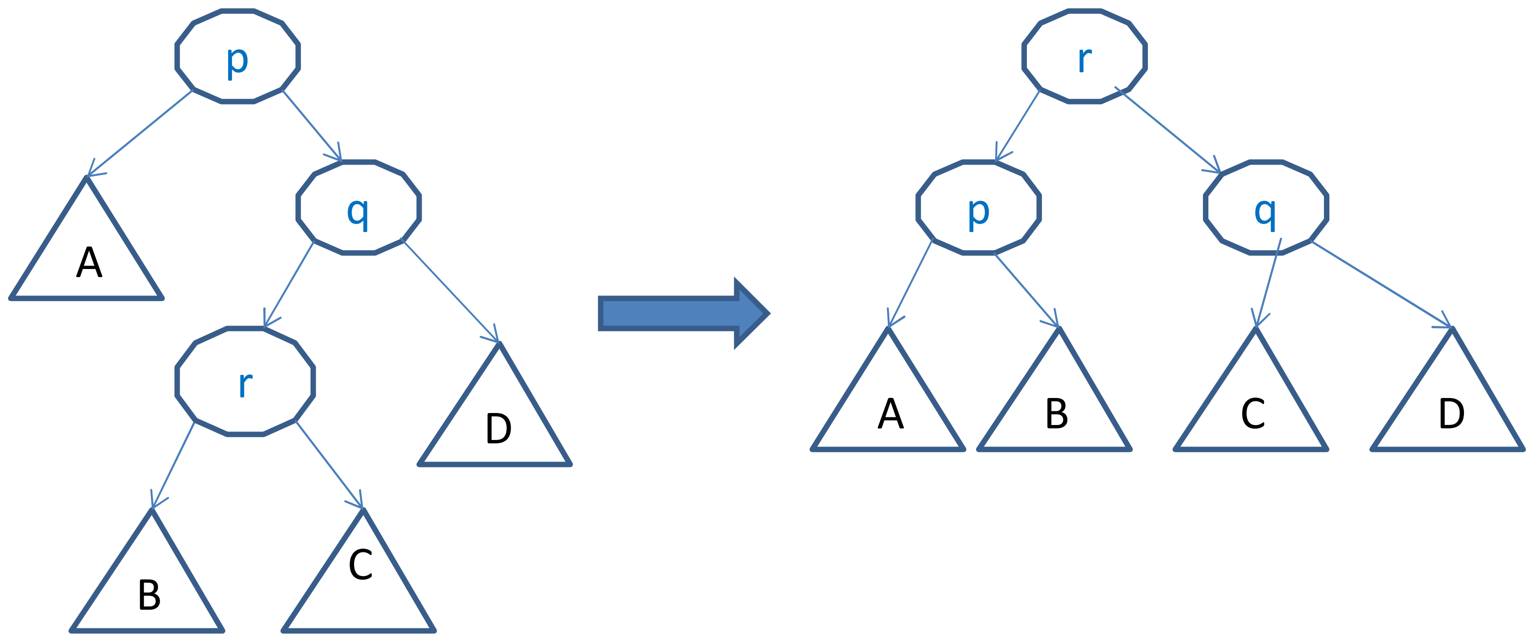
Single Rotation



- Some times, this kind of rotation will not be able to fix the height balancing problem.

Rotations in AVL tree

Double Rotation



- A similar double rotation involves left child's right child and their subtrees.