C, C++, DSA in depth

Binary Tree

Saurabh Shukla (MySirG)

Agenda

1. Binary Tree
2. Complete Binary Tree
3. Almost Complete Binary tree
4. Strict Binary Tree
5. Representation of Binary tree

Binary Tree

A binary tree is defined as a finite set of elements, called nodes, such that

- T is empty (called the Null tree or empty tree), or
- T contains a distinguished node R, called the root of T, and the remaining nodes of T form an ordered pair of disjoint binary trees T_1 and T_2

Any node in the binary tree has either 0, 1 or 2 child nodes.

Complete Binary Tree

All levels are completely filled.

Almost Complete Binary Tree

All levels are completely filled, except possibily the last level and nodes in the last level are all left aligned.

Strict Binary Tree

Each node of a strict Binary Tree will have either 0 or 2 children.

Full Binary Tree

Representation of Binary Tree

There are two possible representations of binary tree

1. Array Representation
2. Linked Representation (by default)

Array Representation

0	1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8	9

Heap

Linked Representation

node

Root

- root is a node pointer
- when root contains NULL, tree is empty.

root
left item right
left item right
left item right
x
left item right
left item right
left item right

Discuss

- How to insert an item in a BT?
- How to traverse a BT?

4	2	5	1	6	3	7

```
void f1(node *root)
{ if(root)
  {
    f1(root->left);
    cout << root->item;
    f1(root->right);
  }
}
```

Left Subtree root Right Subtree

Inorder traversing

```
void f1(node *root)
{
    if(root)
    {
        cout<< root->item;
        f1(root->left);
        f1(root->right);
    }
}
```

preorder traversing

root left subtree right subtree

1 2 4 5 3 6 7

```
void f1(node *root)
{
    if(root)
    {
        f1(root->left);
        f1(root->right);
        cout<<root->item;
    }
}
```

post order

left subtree | right subtree | root

4 5 2 6 7 3 1

How to insert in BT?

There no sepecific rule for insertion of a new node in BT

Binary Search Tree

Duplicate values are not allowed

Example

root
left item right
50
left item right
20
left item right
70
left item right
60
left item right
65