

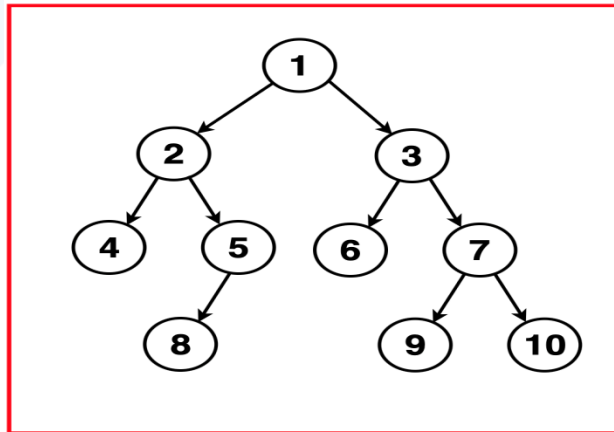
مقرر الخوارزميات و بنى المعطيات 1

جلسة العملي السادسة

(الفصل الثاني 2024-2025)

الأشجار الثنائية

Binary tree traversal



In-order traversal: 4 2 8 5 1 6 3 9 7 10

Pre-order traversal : 1 2 4 5 8 3 6 7 9 10

Post-order traversal : 4 8 5 2 6 9 10 7 3 1

تمثيل بنية الأشجار بلغة ++c

➤ تعريف الصف الممثل لعقد الشجرة:

```
class Node{  
public:  
int data;  
Node *left, *right;  
  
Node(){  
left=right=NULL;  
}
```

```
Node (int dat, Node *lft=NULL, Node *rgt=NULL){  
data = dat;  
left = lft;  
right = rgt;  
}  
};
```

➤ تعريف الصف الممثل للشجرة الثنائية:

```
class BinaryTree{  
public:  
Node *root;  
  
BinaryTree (){  
root = NULL;  
}
```

➤ تابع مسح عقد الشجرة بشكل :inorder

```
void inorder(Node *tree)
{
    if(tree!=NULL)
    {
        inorder(tree->left);
        cout<<tree->data<<" ";
        inorder(tree->right);
    }
}
```

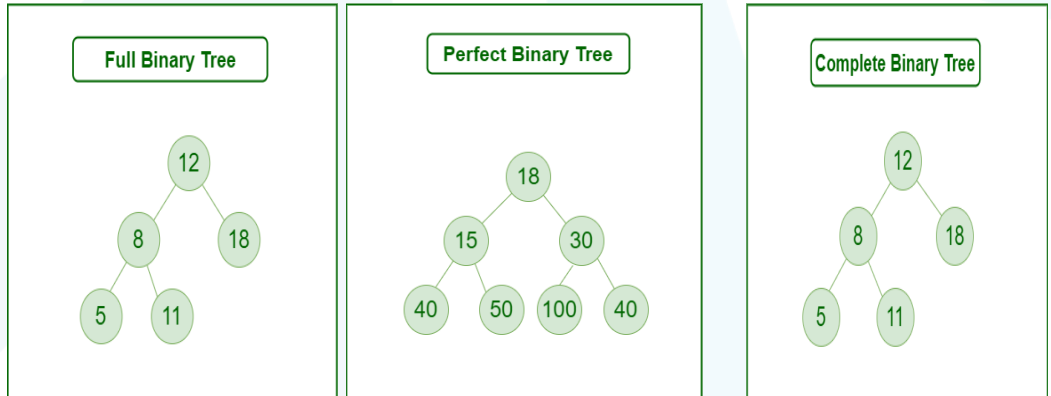
➤ تابع مسح عقد الشجرة بشكل :preorder

```
void preorder(Node *tree)
{
    if(tree!=NULL)
    {
        cout<<tree->data<<" ";
        preorder(tree->left);
        preorder(tree->right);
    }
}
```

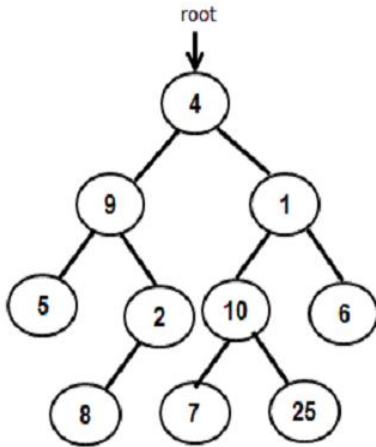
➤ تابع مسح عقد الشجرة بشكل postorder:

```
void postorder(Node *tree)
{
    if(tree!=NULL)
    {
        postorder(tree->left);
        postorder(tree->right);
        cout<<tree->data<<" ";
    }
}
```

Binary Tree Types



أوجد خرج المقطع البرمجي المطبق على الشجرة التالية :

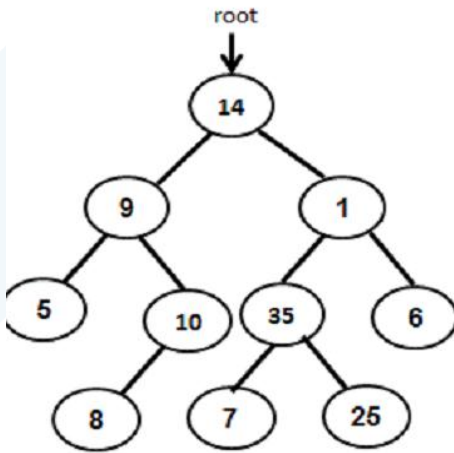


```
Node * p=root;
while(p!=NULL)
{
    cout<< p->data;
    if(p->data%2==0)
        p=p->right;
    else
        p=p->left;
}
```

الخرج :

4 1 10 25

أوجد خرج المقطع البرمجي المطبق على الشجرة التالية :

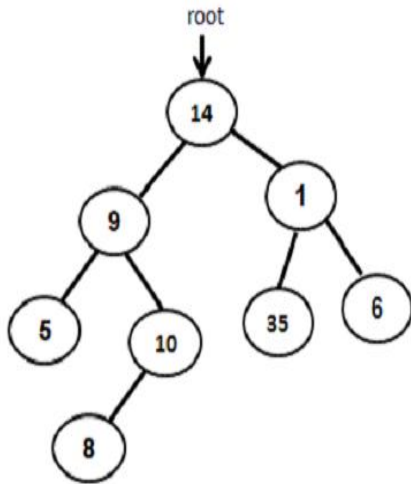


```
Node * ptr=root;
int s=0;
while(ptr!=NULL)
{
    s=s + ptr->data;
    if(ptr->data>10)
        ptr=ptr->left;
    else
        ptr=ptr->right;
}
cout<<s;
```

الخرج

33

أوجد خرج المقطع البرمجي المطبق على الشجرة التالية :



```
Node * p=root->left;
while(p->right!=NULL)
{
    cout<< p->data;
    p=p->right;
}
```

الخرج

9