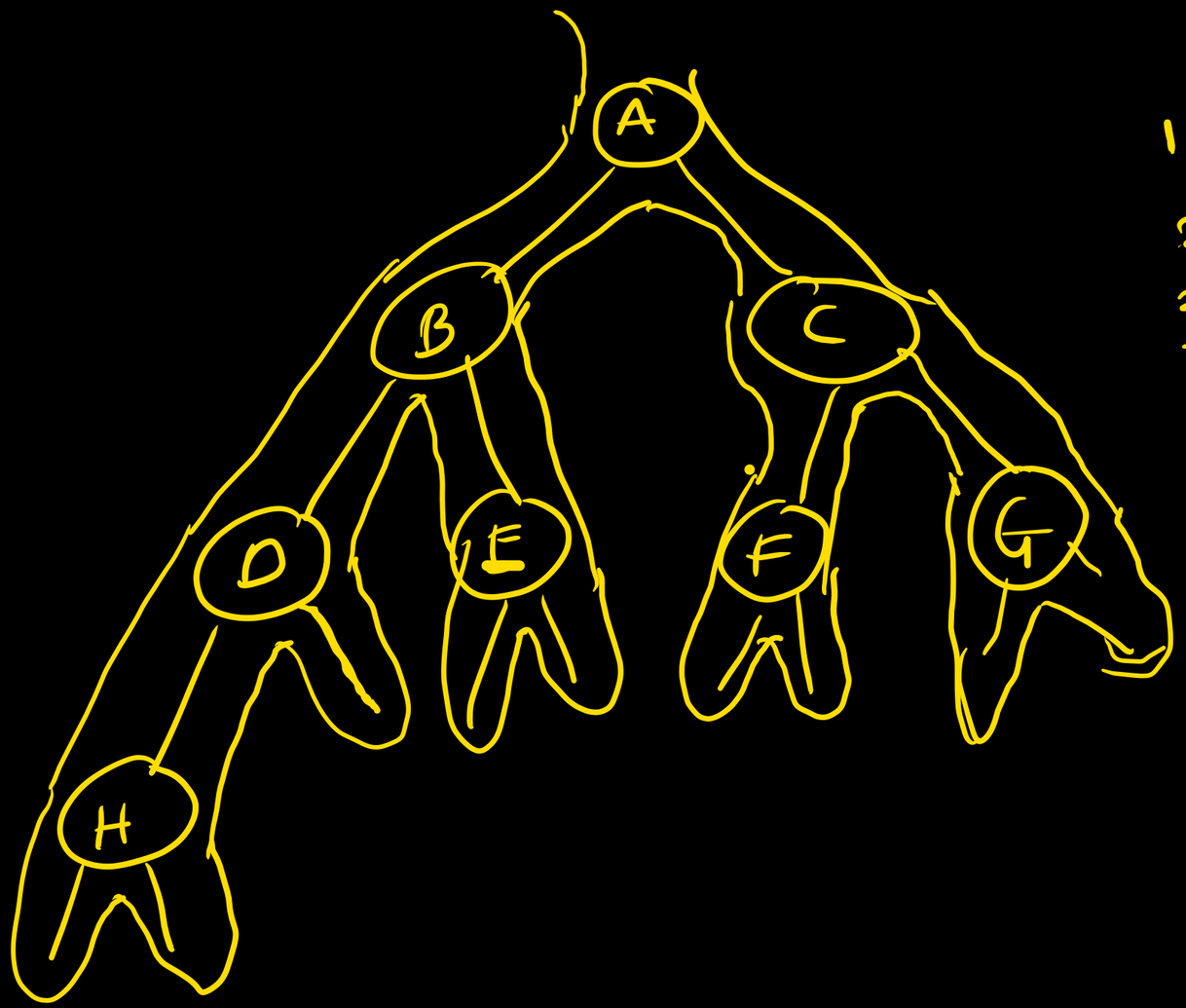
Pre — Root L R

In — L Root R

Post → L R Root



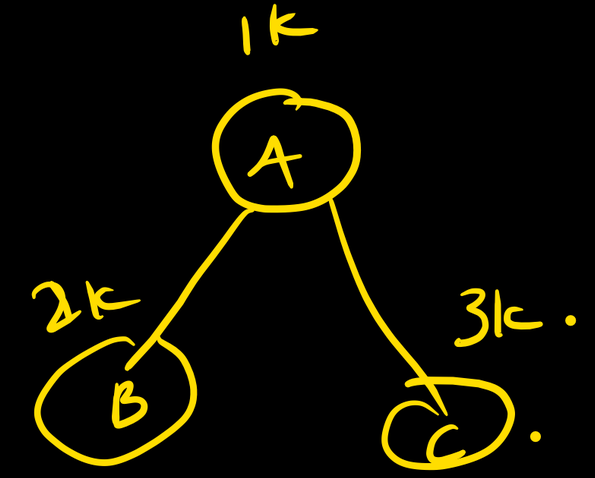
1 Pre - A B D H E C F G
2 In - H D B E A F C G
3 Post - H D E B F G C A.

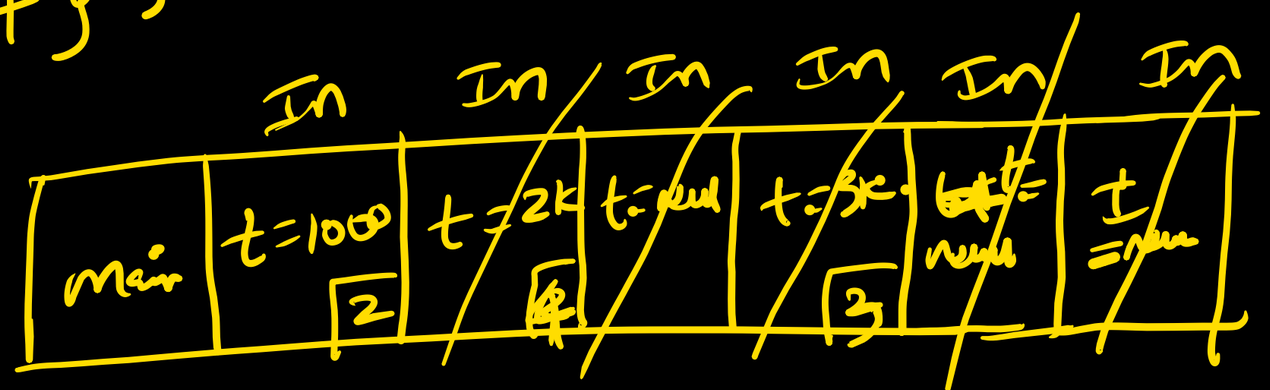
```
void inorder(struct node *t)
{
  if(t!=NULL)
  {
    Inorder(t->left);
      Print("%d", t->data)
    Inorder(t->right
  }
}
```

```
Preorder
{
  Pf
  Pre(L)
  Pre(R)
}
```

```
Postorder
{
  Post(L)
  Post(R)
  Pf
}
```

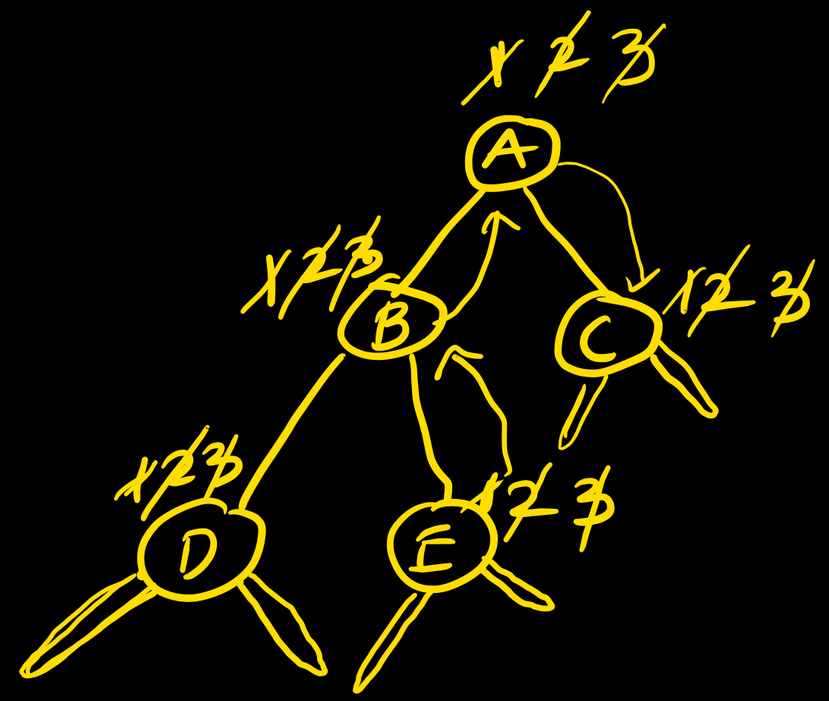
```
In(t)
{ if(t)
 1. In(left)
 2 Pf(data)
 3 In(right)
4 }
```





B C A.

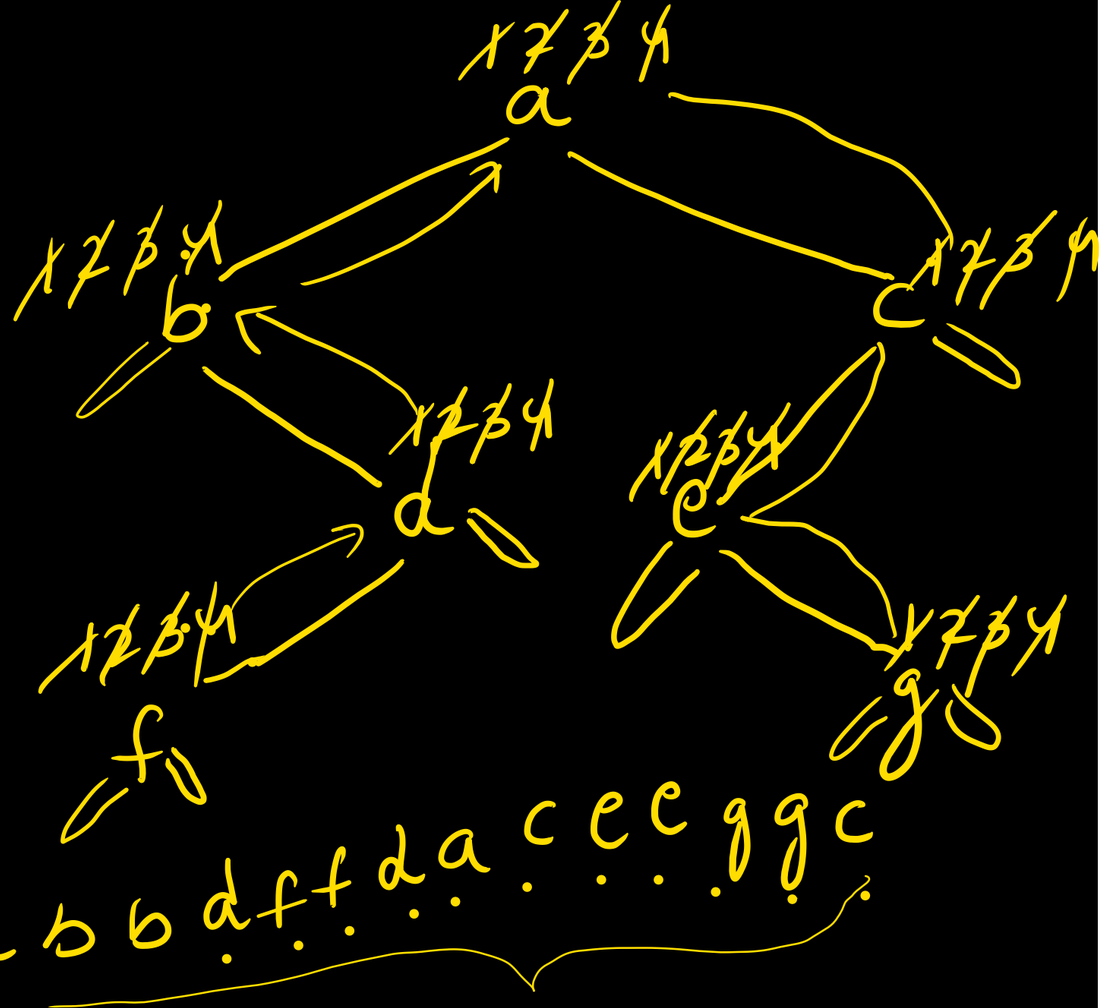
```
InOrder(t)
{
  if(t)
  {
    InOrder(t->left)
    pf(data)
    InOrder(t->right)
  }
}
```



D B E A C

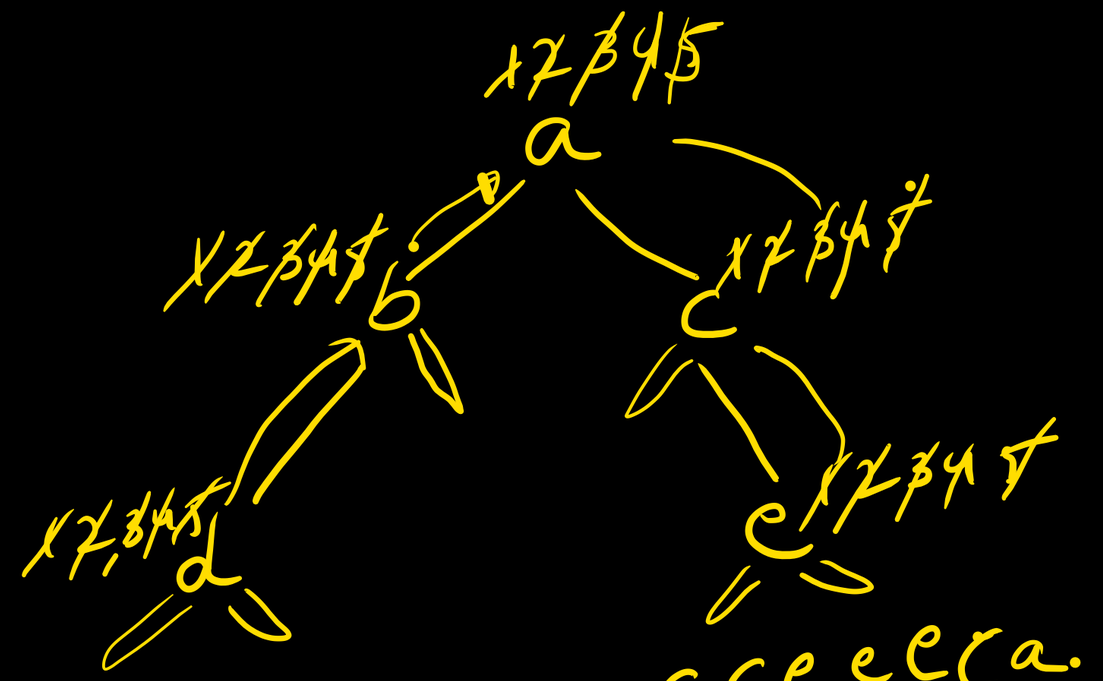
Gate. DO(t) Double Order.

```
{
  if(t)
  {
    1. Pf(t->data)
    2. DO(t->left)
    3 Pf(t->data)
    4. DO(t->right)
  }
}
```



DO(root).

```
TO ( struct node *t)
{
    if (t)
    {
1.    pf (data);
2.    TO (T→left);
3.    pf (data);
4.    TO (T→ right);
      pf (data);
    }
}
```



a b d d d b b a c c e e e c c a.