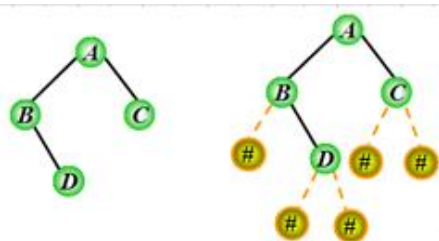
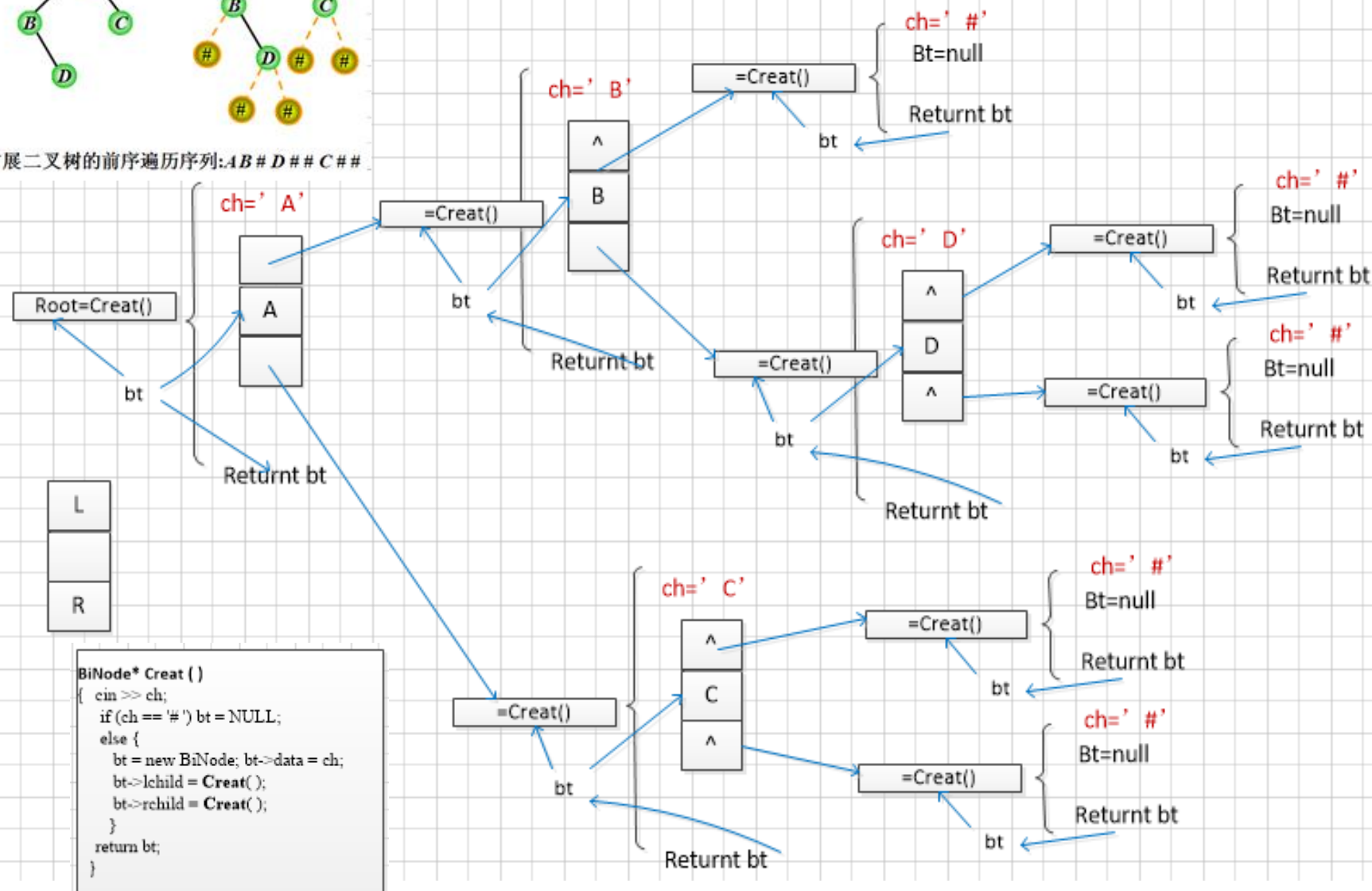


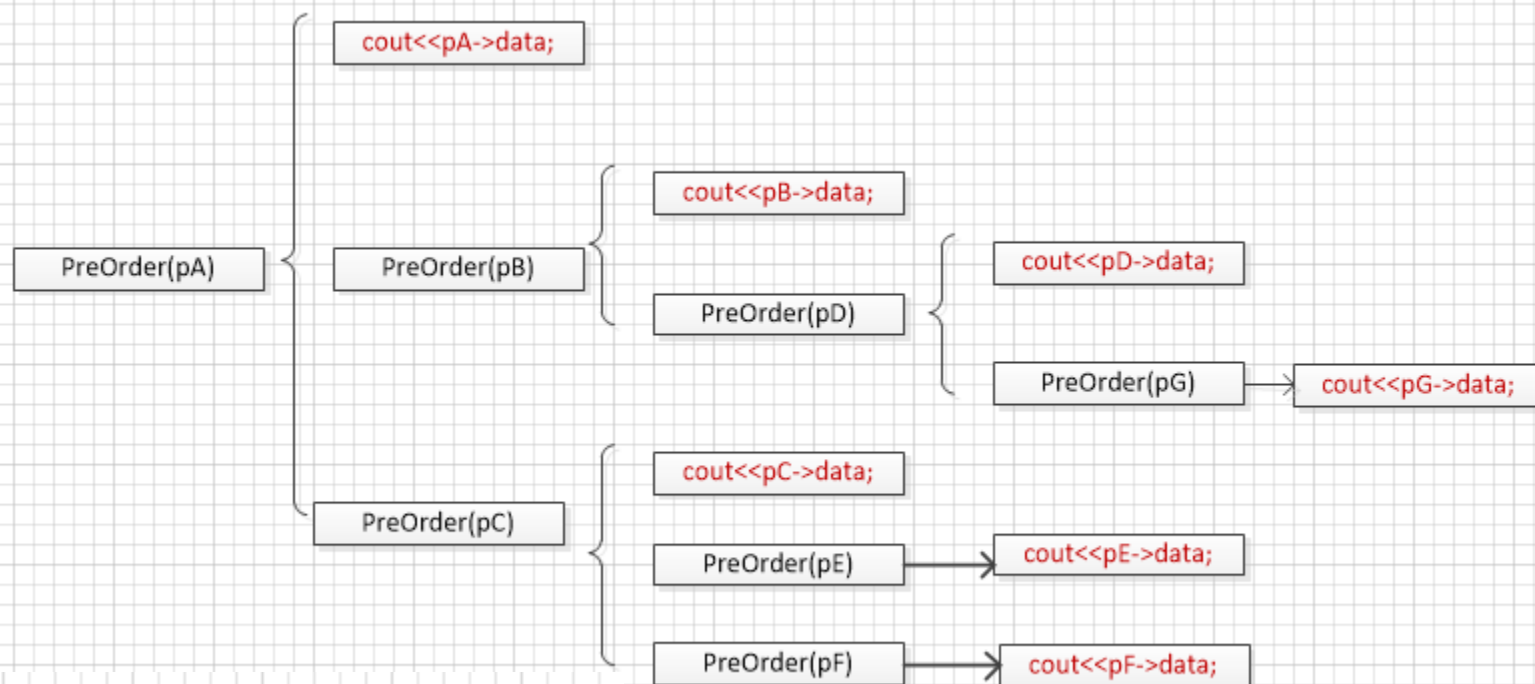
理解递归-前序建二叉树



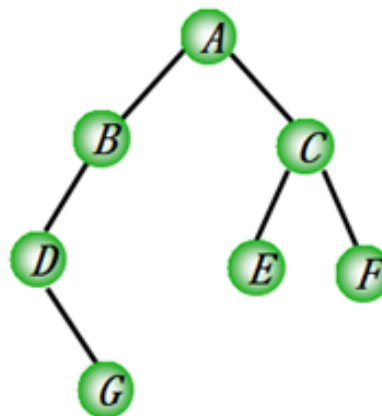
扩展二叉树的前序遍历序列: $AB\#D\#\#C\#\#$



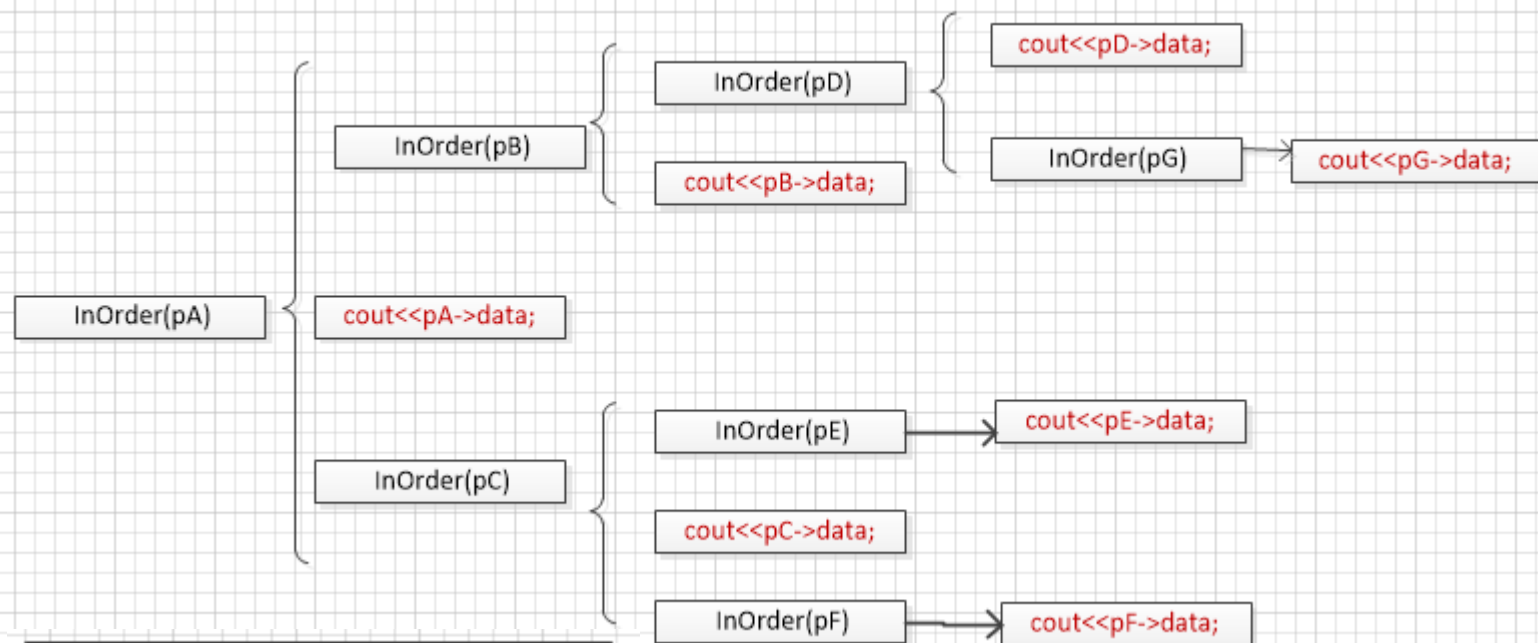
理解递归-前序



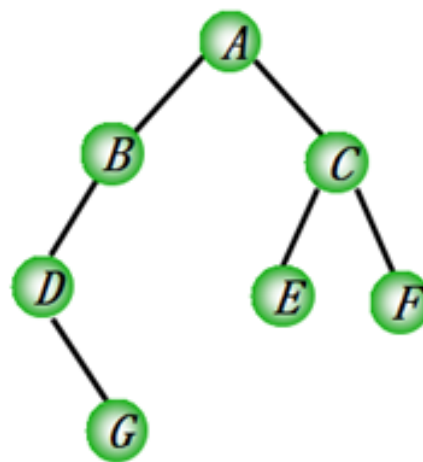
```
void PreOrder(TreeNode *root);  
{ if root==Null  
  return  
  cout<<root->data;  
  if root->Child1!=Null  
    PreOrder( root->Child1);  
  if root->Child2!=Null  
    PreOrder( root->Child2);  
}
```



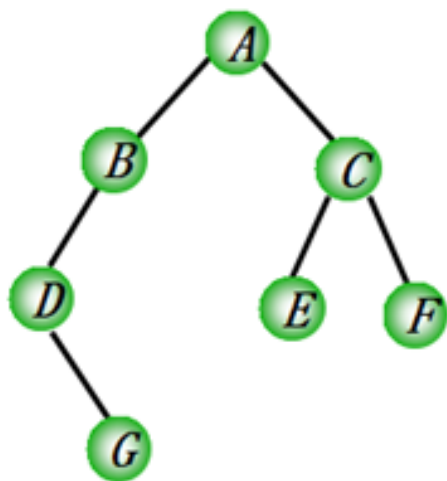
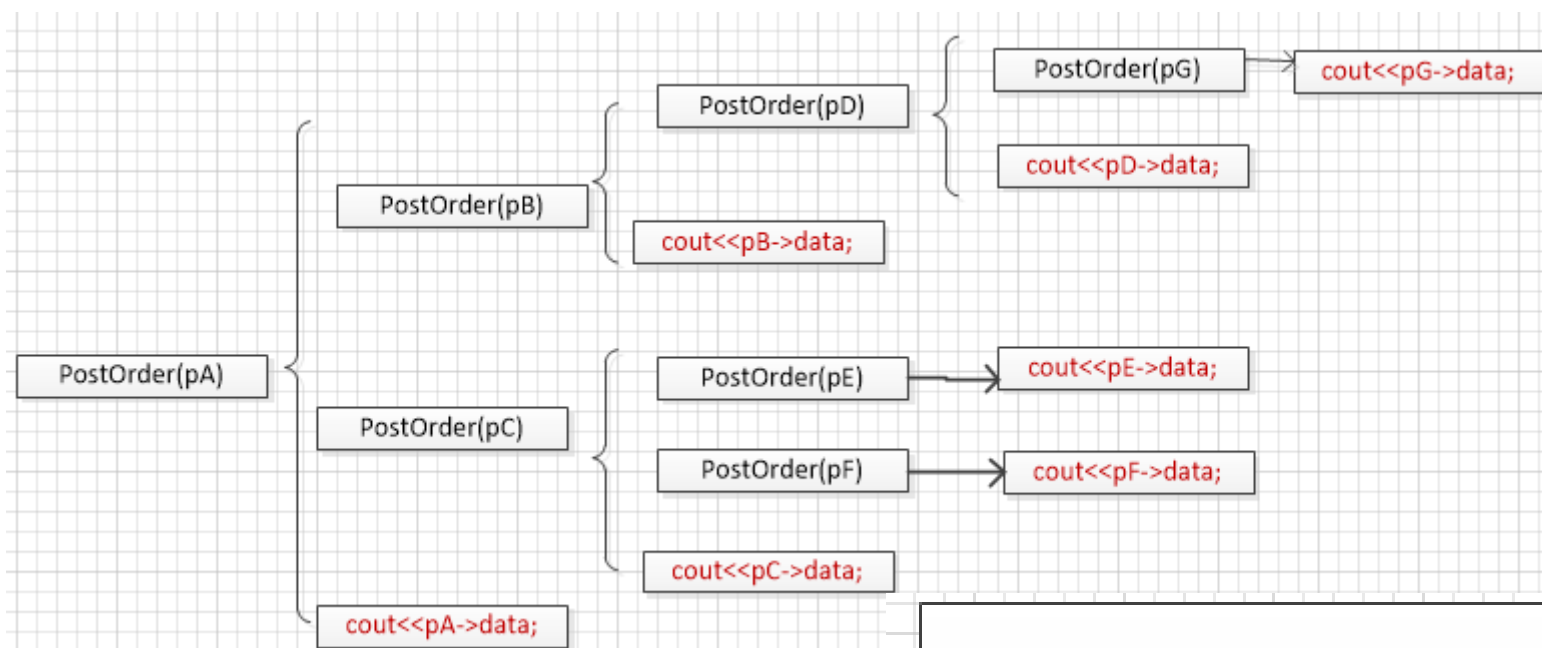
理解递归-中序



```
void InOrder(TreeNode *root);  
{ if root==Null  
  return  
  if root->Child1!=Null  
    InOrder( root->Child1);  
  
  cout<<root->data;  
  
  if root->Child2!=Null  
    InOrder( root->Child2);  
}
```



理解递归-后序

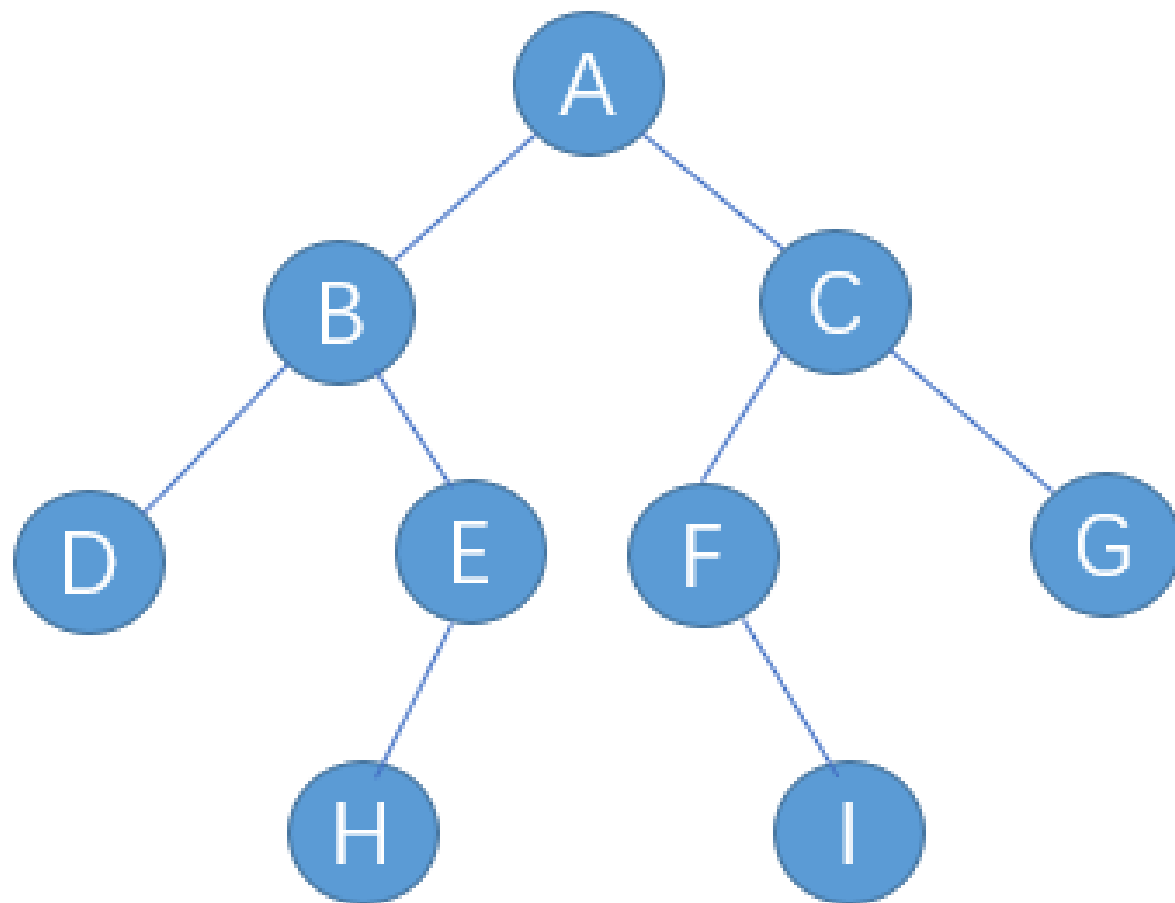


```
Void PostOrder(TreeNode *root);
{ if root==Null
  return
  if root->Child1!=Null
    PostOrder( root->Child1);
  if root->Child2!=Null
    PostOrder( root->Child2);
  cout<<root->data;
}
```

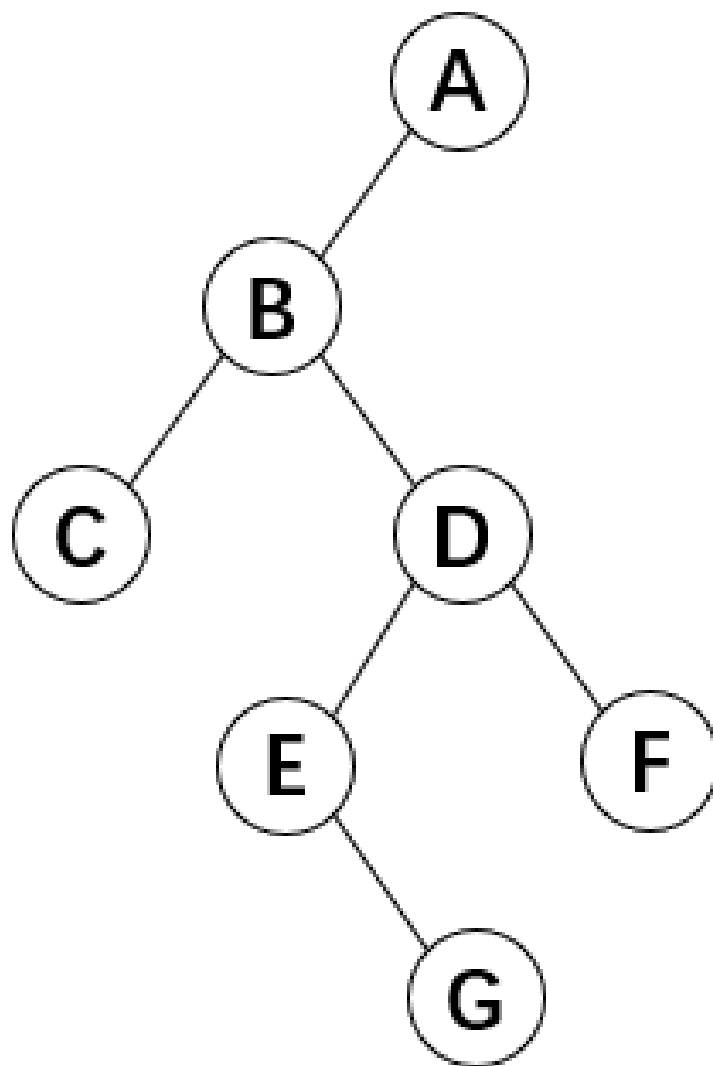
实验内容 1

1. 任选一棵二叉树，使用扩展二叉树的前序遍历序列构造二叉树
2. 画出递归版遍历图和结果，任选一：
前序遍历图及结果；
或 中序遍历图及结果；
或 后序遍历图及结果

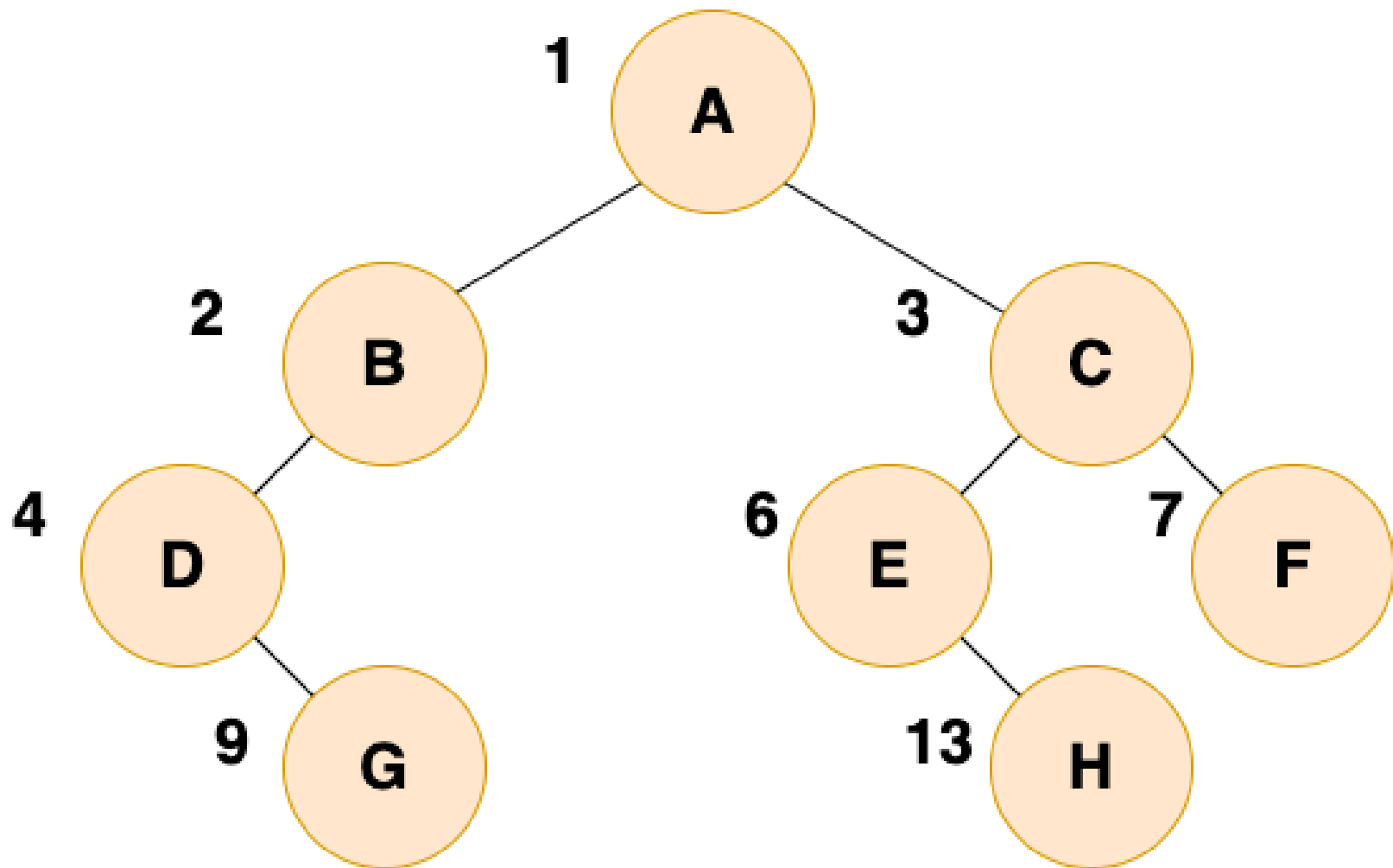
测试例一



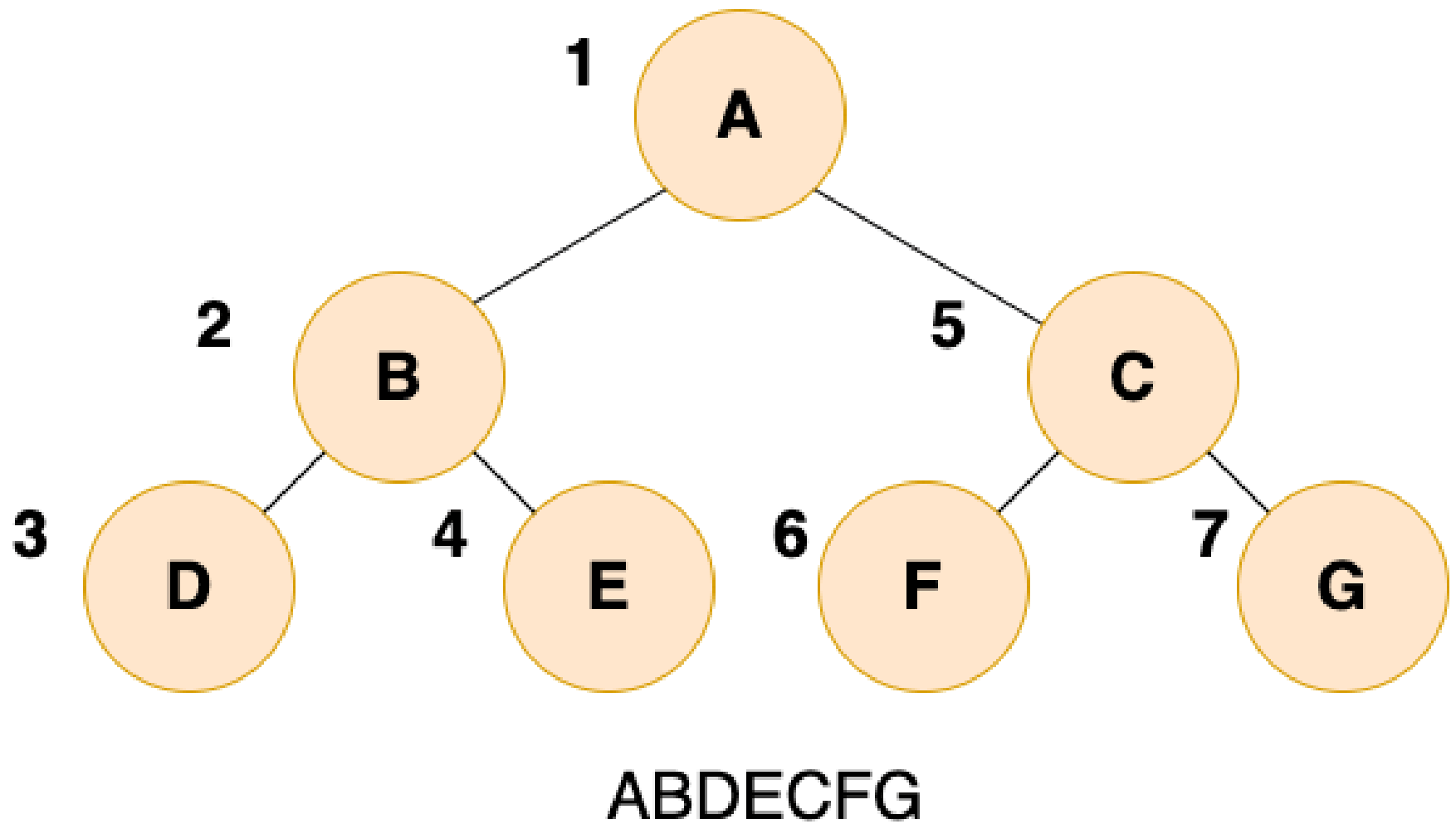
测试例二



测试例三



测试例四



实验内容 2

1. 参考二叉树算法设计文件，实现二叉树相关算法代码（任选二），并用已构造的二叉树进行测试