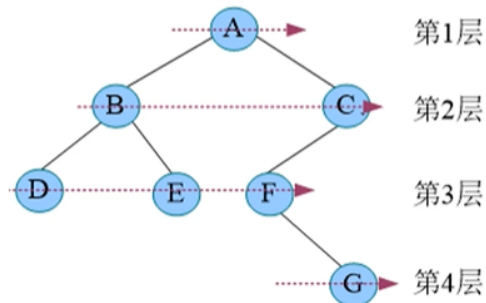
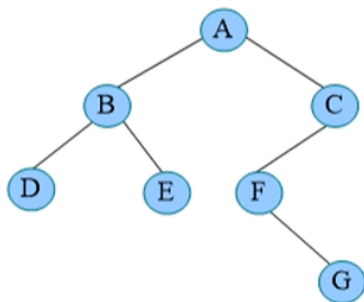


层次遍历，即按照层次的顺序从左向右进行遍历。

**层次遍历秘籍：**首先遍历第1层，然后第2层……同一层按照从左向右的顺序访问，直到最后一层。



通过观察可以发现，先被访问的节点，其孩子也先被访问，先来先服务，因此可以用队列实现。使用队列对下面二叉树进行层次遍历。



遍历方法：

树根入队，判断如果队列不空队头出队，出队的同时判断有无左右孩子，有左右孩子的左右孩子入队……循环直至队列空了结束

--代码实现-->

```
bool Leveltraverse(Btree T){
    Btree p;
    if(!T)
        return false;
    queue<Btree>Q; // 创建一个队列，指针类型
    Q.push(T); // 根指针入队
    while(!Q.empty()){ // 如果队列不空
        p=Q.front(); // 取出队头元素
        Q.pop(); // 队头元素出队
        cout<<p->data<<" ";
        if(p->lchild)
            Q.push(p->lchild); // 左孩子指针入队
        if(p->rchild)
            Q.push(p->rchild); // 右孩子指针入队
    }
    return true;
}
```

