

## BFS 模板

需要确定当前遍历到了哪一层，BFS 模板如下：

```
level = 0           //记录到哪一层了。
while queue != NULL {
    size = queue.size //当前层的节点个数，要全部处理完
    for i in size {
        cur = 出队
        //处理当前节点
        if 当前相邻的节点存在且未被使用把当前节点连接的节点入队
            queue.push()
    }
    //处理当前层
}
```

二叉树的层序遍历：

```
q.push(root);           //先处理根节点
while(!q.empty()) {
    vector<int> level;
    int size = q.size();
    for(int i = 0; i < size; ++i) {
        //当前节点出队列,并处理
        TreeNode *cur = q.front(); q.pop();
        level.push_back(cur->val);
        //当前节点的相关节点入队列
        if (cur->left) q.push(cur->left);
        if (cur->right) q.push(cur->right);
    }
    //将当前层的结果添加到最终结果
    res.push_back(level);
}
```

前提

- 1.数据是有序的
- 2.存在上下边界
- 3.通过index访问

## 二分查找模板

```
left = 0, right = len(array) - 1 //定义左右边界的索引
while(left <= right) {           //终止条件, 当查找的目标值为最右边的值时, left = right
    mid = (left + right) / 2
    if (target == array[mid])
        //find return
    elseif (target > array[mid]) { //目标值比中间值大则修改左边界位置
        left = mid + 1
    }else {
        right = mid - 1
    }
}
```