**二叉树的层次遍历**

一、

1. 层次遍历其实就只是按每一层的从左往右进行的遍历方式，那么我们就需要用一个队列，利用队列先进先出的特性来进行访问
2. 代码

//层次遍历

#include <queue>

void levelOrder\_tree(BiTree \*T) {

queue<BiTree \*> q;

q.push(T); //先把我们的投节点的元素先推进去队列中去

while (!q.empty()) {

//如果不为空，那么我们pop出来，打印处理，把其的左右孩子一次入队 BiTree\* p = q.front(); //取出队列的对头元素

q.pop(); //取完之后把对头的元素弹出来

cout << p->data << " "; //打印出来

//如果有左孩子，就把左孩子推进去

if (p->left\_child != NULL) {

q.push(p->left\_child);

}

if (p->right\_child != NULL) {

q.push(p->right\_child);

}

}

}