用数组实现树层序遍历问题

代表例题：

1099 **Build A Binary Search Tree

代码实现：

```
#include<iostream>
#include <algorithm>
#include <vector>
using namespace std;
class node
{
public:
    int data, right, left, in, level;
    node()
    {
        this->data = this->in = this->level = -1;
        this->right = this->left = -1;
    }
};
vector<node> tree;
vector<bool> visit;
vector<int> seq;
int number;
int init()
{
    cin >> number;
    tree.resize(number);
    visit.assign(number, false);
    seq.assign(number, 0);
    for(int i=0;i<number;i++)
    {
        cin >> tree[i].left >> tree[i].right;
        if (tree[i].left != -1)
            visit[tree[i].left] = true;
        if (tree[i].right != -1)
            visit[tree[i].right] = true;
    }
    for(int i=0;i<number;i++)
    {
        cin >> seq[i];
    }
    sort(seq.begin(), seq.end());
    return 0;
}
int index = 0;
void DFS(int root,int in,int level)
{
    if (root == -1)
        return;
```

```
    DFS(tree[root].left, in * 2,level+1);
    tree[root].in = in;
    tree[root].data = seq[index++];
    tree[root].level = level;
    DFS(tree[root].right, in * 2 + 1, level + 1);
}
bool cmp(node a,node b)
{
    if (a.level != b.level)
        return a.level < b.level;
    else
        return a.in < b.in;
}
int main()
{
    int root = init();
    DFS(root, 1,1);
    sort(tree.begin(), tree.end(), cmp);
    cout << tree[0].data;
    for(int i=1;i<number;i++)
    {
        cout << " " << tree[i].data;
    }
    return 0;
}
```