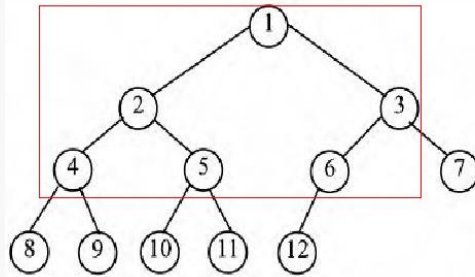


完全二叉树非叶子部分后序遍历

题目描述： 给定一个以顺序储存结构存储整数值的完全二叉树序列（最多1000个整数），请找出此完全二叉树的所有非叶子节点部分，然后采用后序遍历方式将此部分树（不包含叶子）输出。

- 1、只有一个节点的树，此节点认定为根节点（非叶子）。
- 2、此完全二叉树并非满二叉树，可能存在倒数第二层出现叶子或者无右叶子的情况



其他说明：二叉树的后序遍历是基于根来说的，遍历顺序为：左-右-根

输入描述：
一个通过空格分割的整数序列字符串

输出描述：
非叶子部分树结构的后序遍历结果

补充说明：输出数字以空格分隔

示例1

输入：1 2 3 4 5 6 7

输出：2 3 1

说明：找到非叶子部分树结构，然后采用后续遍历输出

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 10005;

int a[maxn];

bool isleaf(int now) {
    if (now * 2 > a[0]) return true; else return false;
}

void dfs(int now) {
    if (!isleaf(now * 2)) dfs(now * 2);
    if (!isleaf(now * 2 + 1)) dfs(now * 2 + 1);
    cout << a[now] << ' ';
}

int main() {
    int x;
    a[0] = 0;
    while (cin >> x) {
        a[++a[0]] = x;
    }
}
```

```
    dfs(1);  
    return 0;  
}
```