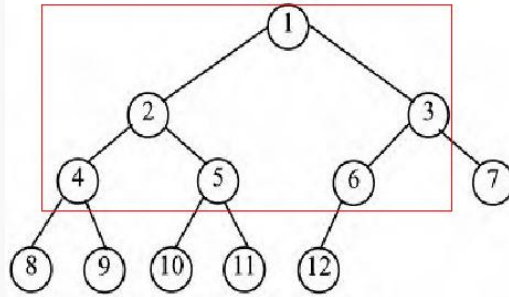


题目描述: 给定一个以顺序储存结构存储整数值的完全二叉树序列 (最多1000个整数), 请找出此完全二叉树的所有非叶子节点部分, 然后采用后序遍历方式将此部分树 (不包含叶子) 输出。

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

// 本题为考试单行多行输入输出规范示例, 无需提交, 不计分。

```
#include <iostream>
```

```
using namespace std;
```

```
const int N = 10000;
```

```
int tree[N];
```

```
void print(int index, int n, bool &f) {
```

```
    if (2*index+1<n) {  
        print(2*index+1,n,f);
```

```
    }
```

```
    if (2*index+2<n) {  
        print(2*index+2,n,f);
```

```
    }
```

```
    if(f) {  
        std::cout << " ";
```

```
    }
```

```
    std::cout<<tree[index];
```

```
        f=true;
    }
    int main() {
        int node;
        int top = 0;
        while(cin >> node){
            tree[top++] = node;
        }
        bool f = false;
        print(0, top/2, f);
        return 0;
    }
```