

## 一、编程题

### ACM:寻找最富裕的小家庭

#### 题目描述:

在一棵树中，每个节点代表一个家庭成员，节点的数字表示其个人的财富值，一个节点及其直接相连的子节点被定义为一个大家庭。

现给你一棵树，请计算出最富裕的小家庭的财富和。

#### 输入描述:

第一行为一个数  $N$ ，表示成员总数，成员编号  $1-N$ ， $1 \leq N \leq 1000$

第二行为  $N$  个空格分隔的数，表示编号  $1-N$  的成员的财富值。 $0 \leq \text{财富值} \leq 1000000$

接下来  $N-1$  行，每行两个空格分隔的整数( $N1, N2$ )，表示  $N1$  是  $N2$  的父节点。

#### 输出描述:

最富裕的小家庭的财富和

#### 补充说明:

##### 示例1

输入：4

100 200 300 500

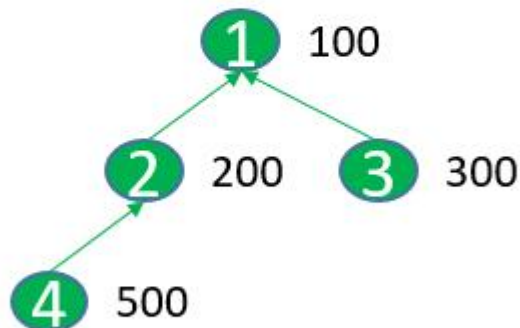
1 2

1 3

2 4

输出：700

说明：



成员1,2,3组成的小家庭财富值为600

成员2,4组成的小家庭财富值为700

示例2

输入：4

100 200 300 500

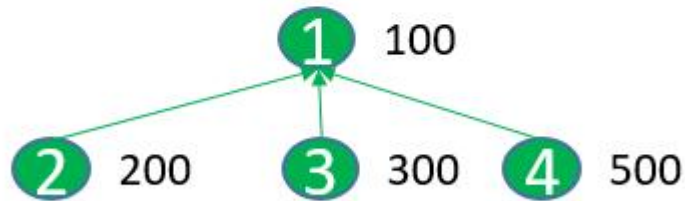
1 2

1 3

1 4

输出：1100

说明：



成员1,2,3,4组成的小家庭财富值为1100

代码：

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;
```

// 注意类名必须为 Main, 不要有任何 package xxx 信息

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner in = new Scanner(System.in);
```

```
        // 注意 hasNext 和 hasNextLine 的区别
```

```
        while (in.hasNext()) { // 注意 while 处理多个 case
```

```
            int input = in.nextInt();
```

```
            Map<Integer, Integer> moneyMap = new HashMap<>();
```

```
            for (int i = 0; i < input; i++) {
```

```
                moneyMap.put(i + 1, in.nextInt());
```

```
            }
```

```
            Map<Integer, List<Integer>> map = new HashMap<>();
```

```
            for (int i = 0; i < input - 1; i++) {
```

```
                int key = in.nextInt();
```

```

        List<Integer> list;
        if (map.containsKey(key)) {
            list = map.get(key);
            list.add(in.nextInt());
        } else {
            list = new ArrayList<>();
            list.add(key);
            list.add(in.nextInt());
            map.put(key, list);
        }
    }

    int max = 0;
    for (List<Integer> members : map.values()) {
        int allMoney = 0;
        for (Integer member : members) {
            allMoney += moneyMap.get(member);
        }
        max = Math.max(max, allMoney);
    }

    System.out.println(max);
}
}
}

```