

# Java方向编程题答案

## day5

[编程题]23630-神奇的口袋

链接: <https://www.nowcoder.com/questionTerminal/9aaea0b82623466a8b29a9f1a00b5d35>

【题目解析】：大家以后看到这种一堆文字的题目一定不要害怕，多读几遍题大家就会发现，其实这道题就是一个高中就学过的排列组合问题，这类问题使用递归就可以解决。

【解题思路】：采用递归思想：①物品n个，物品体积逐一放入weight[]中 ②递归函数count(int s,int n)：其中s为剩余物品重量，n为剩余可选的物品个数

则分以下两步递归求解：

a.从后往前装，装上weight[n]后，若剩余物品仍然有解

则count(s-weight[n],n-1);

b.若装了weight[n]后，无解，则删除该包,尝试第n-1个

count(s,n-1);

【示例代码】：

```
import java.util.*;
public class Main {
    static int[] weight;
    static int N;
    static int count=0;
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        while (input.hasNext()) {
            N = input.nextInt();
            weight = new int[N+1];

            for (int i = 1; i <= N; i++) {
                weight[i] = input.nextInt();
            }

            count(40,N);
            System.out.println(count);
        }
    }
    public static void count(int s,int n) {
        //如果正好装满
        if(s==0) {
            ++count;
            return ;
        }
        //是s<0或n<1则不能完成
        if(s<0 || (s>0&& n<1))
```

```
        return ;
    count(s-weight[n],n-1);
    count(s,n-1);
}
}
```

[编程题]23281-用两个栈实现队列

链接<https://www.nowcoder.com/questionTerminal/54275ddae22f475981afa2244dd448c6>

【题目解析】：

话不多说，当时讲栈的课后练习题之一，经典问题，当时没做的童鞋一定现在要做，当时完成的童鞋再回顾下自己当时的思路，你能写出更简单思路更清晰的代码吗？

【解题思路】：

- 栈1用来作入队列
- 栈2用来出队列，当栈2为空时，栈1全部出栈到栈2,栈2再出栈（即出队列）

【示例代码】：

```
import java.util.Stack;
public class Solution {
    Stack<Integer> stack1 = new Stack<Integer>();
    Stack<Integer> stack2 = new Stack<Integer>();
    public void push(int node) {
        stack1.push(node);
    }
    public int pop() {
        if(stack1.empty() && stack2.empty()){
            throw new RuntimeException("Queue is empty!");
        }
        if(stack2.empty()){
            while(!stack1.empty()){
                stack2.push(stack1.pop());
            }
        }
        return stack2.pop();
    }
}
```