**7-27 盲盒包装流水线**

// 我的思路：字符串映射int，再通过int在数组里获得编号，二维数组模拟栈存类型stirng->int->int

// 题解思路：用string数组维护编号与字符串，再套入map里,一维数组模拟二维数组+模拟栈

// 我的思路并没有问题，只是由于第一个读入的字符串在map中的值被映射为了0，会被判断不存在。输出Wrong Number

// 因此只要将判断Worng Number的条件改一下，写成mp.conut(num)即可，这样就可以避免合法元素map映射为0与非法元素混淆的问题

// 再就是二维数组的问题，看错了题目关键信息，盲盒总量 N（≤10^ 5），范围太大，不能用二维数组。

// 还有就是本题的数组模拟栈，只要读入时将下标倒转即可，

// 题解的思路就比较精妙，用string数组来维护编号和字符串，再通过编号直接将string与徽章类型通过map联系在一起，直接通过string来访问到map，这样编号取多少都没影响

// 不过还是用m.conut(num)这个函数来判断值有没有存入过map比较保险，以后遇到用map记录元素出现情况的话就用m.conut(num);

二刷卡壳

// 先看数据范围，1e5,100,1e4，

// 总量是1e5，查询1e4次，肯定不能双循环，可能是预处理o（1）查询

// 货栈看到**栈**，要打起12分精神，储存方式必较特殊

// 当货栈为空时，打包机会暂停，等待下一批徽章压入货栈。分析储存顺序

// 徽章分批进入栈，从栈顶弹出徽章来装入盲盒，弹完后再压新栈，所以

// 用例输入里1 2 3 4 5 9 8 7 6 1的徽章顺序，实际上是

// 5 4 3 2 1 1 6 7 8 9 的装盒顺序通过双循环i \* 5**(货栈容量，一行数量)/写错了导致卡壳** + 倒序j来实现

// 用数组维护装盒顺序与对应勋章，还有装盒顺序与编号。通过这顺序将编号与勋章建立联系，实现string到徽章o1查询

// 还要思考用什么容器来维护信息

// 装盒顺序与编号可以用string数组或map，

// 最终的联系则由于具有string查询徽章的特征，只能用map

// string数组的速度是优于map的，但是o1的复杂度并无大碍

// 之前string数组用的少，这次用用

**// 卡壳[Error] invalid types ‘int[int]‘ for array subscript**

**// no match for ‘operator[]’ (operand types are ‘int’ and ‘std::\_\_cxx11::string {aka std::\_\_cxx11::basic\_string<char>}’)**

**// 都是变量重复定义导致出现了int【int】这种出乎意料的组合**

二刷[云剪贴板 - 洛谷 | 计算机科学教育新生态 (luogu.com.cn)](https://www.luogu.com.cn/paste/r1nrngqp)

三刷：换成int+数组完美通过

// 对每个查询编号，在一行中输出该盒子中装的徽章类型。如果编号是错误的，则在一行中输出 Wrong Number。

// 题目要通过编号查询徽章类型，也就是要将编号与徽章类型建立联系，可以想到用map或数组

// 因为编号与徽章是分开读入的，所以先用一个数组或map来储存编号，读入徽章时再拿出来建立联系

// 对于徽章，输入写成了一个矩阵的形式，是出题人的陷阱，暗示你用二维数组来读取，string，通过二维string数组来与徽章建立联系

// 可以试试

// 建立联系后o1查询即可

// 当货栈为空时，打包机会暂停，等待下一批徽章压入货栈。首先徽章通过进货口被压入货栈里，空盒在履带上从左向右传送。每次从货栈里弹出一枚徽章，进入打包机，装入一只空盒，打包后继续向右边传送。当货栈为空时，打包机会暂停，等待下一批徽章压入货栈。

//

// 对于栈，先入的徽章后出，所以应该是倒转的顺序，但是此题的栈有容量，s，徽章以s个为一组倒序装入盒子

// 通过这个可以知道1 2 3 4 5 9 8 7 6 1 会变成 5 4 3 2 1 1 6 7 8 9

[云剪贴板 - 洛谷 | 计算机科学教育新生态 (luogu.com.cn)](https://www.luogu.com.cn/paste/f139vnbr)