

# 第四周实验报告

沈家成

2017 年 10 月 11 日

## 1 1005 数独

### 1.1 走过的弯路

在思考如何检查 1 ~ 9 每个数字都出现的时候，想简单了，认为只要直接每行、每列、每个九宫格的数字相加，看看是不是 45 就可以了。后来想到没那么简单，可能会有一对 1-9 替换 2-8 的情况，但是这样和依然是 45。

### 1.2 解决办法

定义了一个 9 位的布尔型数组，用来记录每个数字是否出现。遍历完一行、一列或一个九宫格后，就检查这个数组是不是全部为 `true`。

### 1.3 小结

为了提高准确率，要充分发掘信息，宽松的必要性条件是不够的，需要严格的必要条件，最好找到充要条件。

## 2 1049 火车调度

### 2.1 主要难点

用到了栈和队列两种数据结构，整个程序很复杂。

### 2.2 解决方法

模块化开发。先测试自己写的栈类和队列类，入栈、出栈、入队、出队等功能都正常运行后，再编写专门用于本题的类。

### 2.3 小结

在编写大型程序的时候不能着急，要及时测试模块运行是否正常。如果到了最后才一起 debug，很难确定 bug 的地点。

## 3 1574 调皮的助教

### 3.1 难点分析

助教和题的关系明显是一个循环的结构。由于需要多次调用，对于出题算法的时间复杂度要求特别高。

最终将会出  $10^{1,000,000,000}$  次题，如果完全依靠计算机计数，将会是指数级的时间复杂度，很可能宕机。

## 3.2 解决方法

### 3.2.1 看待角度

看待问题的角度不同，算法也就不同。用循环的结构来储存数据的时候，把谁看作数据，把谁看作地址；又把谁看作动的，把谁看作静的。一般来说，地址是静的，数据是动的。那么最符合常识的看法是，把助教看作静止的地址，把题看作运动的数据，每次出题，就是每个地址的数据改变。但是这个每次都要涉及数据的大量移动，是  $O(N)$  的时间复杂度。

但是如果换一种看法呢？如果把地址看作动的，把数据看作静止的，那么是每道题对应的助教不同，再把物理地址抽象成逻辑地址，只要知道 0 号助教出第几道题，不就on知道所有其他助教出什么题了吗？所以最终，设置一个  $n$  个元素的数组，里面存放 0  $n-1$  号题，再定义一个变量表示 0 号助教在数组中的位置。每次出题，就是 0 号助教的位置加  $m$ ，通过取余限制在循环范围内。这样，每次只需要进行一次运算，时间复杂度为  $O(1)$ 。

甚至再进一步，0 号助教所处的位置，和出题号在数值上是一致的，因此都不需要设置一个数组，直接用位置就可以表示出题号了。

### 3.2.2 递归计算

递归函数常常会消耗大量资源，但是如果使用递归的思想来进行  $10^{1,000,000,000}$  次循环，就可能将其降到 1,000,000,000 次循环，这是计算机能够解决的问题。

$10^{1,000,000,000}$  次循环，可以看作 10 次  $10^{999,999,999}$  循环……100 次循环，可以看作 10 次 10 循环。而每层循环的结果，可以看作助教的地址又向后移动了几位。因此，只要记录下移动的距离，将每十次移动的最终距离传给上一层，作为上一层每次移动的距离，就实现了递归操作。

即使这样，十次测试中依然有一次超时。因此，可以一次循环里进行三次递归，也就是计算移动一千次后的位置，最后再根据三的余数处理各种情况。

### 3.3 小结

如果一个角度看问题很复杂，换个角度看问题，可能就简单多了。

递归函数经常通过多次运行来得到数值，如果反着想，就可以用数值代表多次运算，从而降低时间复杂度。