

## NOIP2006 普及组解题报告

试题名称	random	happy	count	sequence
目录	random	happy	count	sequence
输入文件名	random.in	happy.in	count.in	sequence.in
输出文件名	random.out	happy.out	count.out	sequence.out
算法	排序（快速排序、起 泡排序等，最优桶排 序）	动态规划+回滚优化 （LINUX 上测不用回 滚，XP 不回滚会造成栈 溢出，程序无法运行）	字符串处理 （读懂题目就好做 了）	递推 （某个点数据有问题， 但用 long 溢出后刚好 得到正确答案）
难度	0	3	2	1
编程复杂度	1	3	3	2

作者：网络虾客

QQ: 793048

E-mail : [normanyahq@yahoo.com.cn](mailto:normanyahq@yahoo.com.cn)

网站 : [www.oiers.cn](http://www.oiers.cn)

广西师范大学附属外国语学校

## 前言

就我的感觉而言，今年的题目比去年容易得多。轻轻松松就可以拿到 400 了。

但是很多高手却只拿到了 390 分。原因是没仔细看清楚要求，用了 `long long (int64)` 的数据类型，结果有一个点错了。很遗憾。实际上 NOIP 是禁止使用 `long long` 的。

从今年的题目可以看出，第一题仍然是明显的送分题。但第二题第三题却明显难于第四题。所以，以后比赛要总结好经验，不要以为前面的题目就是容易的。

## 1. 明明的随机数

(random.pas/c/cpp)

### 【问题描述】

明明想在学校中请一些同学一起做一项问卷调查，为了实验的客观性，他先用计算机生成了  $N$  个 1 到 1000 之间的随机整数 ( $N \leq 100$ )，对于其中重复的数字，只保留一个，把其余相同的数去掉，不同的数对应着不同的学生的学号。然后再把这些数从小到大排序，按照排好的顺序去找同学做调查。请你协助明明完成“去重”与“排序”的工作。

### 【输入文件】

输入文件 random.in 有 2 行，第 1 行为 1 个正整数，表示所生成的随机数的个数：

$N$

第 2 行有  $N$  个用空格隔开的正整数，为所产生的随机数。

### 【输出文件】

输出文件 random.out 也是 2 行，第 1 行为 1 个正整数  $M$ ，表示不相同的随机数的个数。

第 2 行为  $M$  个用空格隔开的正整数，为从小到大排好序的不相同的随机数。

### 【输入样例】

```
10
20 40 32 67 40 20 89 300 400 15
```

### 【输出样例】

```
8
15 20 32 40 67 89 300 400
```

思路解析：

很简单的。排序一般都能过，不过最优算法应该是桶排序吧。这里只介绍桶排序。用一个 1 到 1000 的数组储存每个数的出现情况， $t$  每出现一次， $a[t]=a[t]+1$ ，而且，同时操作  $\text{if}(a[t]>1)$ ，即如果某个数同时出现了 1 次以上，则总数=总数-1。然后从 1 到 1000 的循环，如果  $a[i]\neq 0$ ，输出。 $O(n)$  的效率。

## 2. 开心的金明

(happy.pas/c/cpp)

### 【问题描述】

金明今天很开心，家里购置的新房就要领钥匙了，新房里有一间他自己专用的很宽敞的房间。更让他高兴的是，妈妈昨天对他说：“你的房间需要购买哪些物品，怎么布置，你说了算，只要不超过  $N$  元钱就行”。今天一早金明就开始做预算，但是他想买的东西太多了，肯定会超过妈妈限定的  $N$  元。于是，他把每件物品规定了一个重要度，分为 5 等：用整数 1~5 表示，第 5 等最重要。他还从因特网上查到了每件物品的价格（都是整数元）。他希望在不超过  $N$  元（可以等于  $N$  元）的前提下，使每件物品的价格与重要度的乘积的总和最大。

设第  $j$  件物品的价格为  $v[j]$ ，重要度为  $w[j]$ ，共选中了  $k$  件物品，编号依次为  $j_1, j_2, \dots, j_k$ ，则所求的总和为：

$$v[j_1] * w[j_1] + v[j_2] * w[j_2] + \dots + v[j_k] * w[j_k]。 (其中 * 为乘号)$$

请你帮助金明设计一个满足要求的购物单。

### 【输入文件】

输入文件 happy.in 的第 1 行，为两个正整数，用一个空格隔开：

$N \quad m$

（其中  $N$  ( $< 30000$ ) 表示总钱数， $m$  ( $< 25$ ) 为希望购买物品的个数。）

从第 2 行到第  $m+1$  行，第  $j$  行给出了编号为  $j-1$  的物品的的基本数据，每行有 2 个非负整数

$v \quad p$

（其中  $v$  表示该物品的价格 ( $v \leq 10000$ )， $p$  表示该物品的重要度 (1~5)）

### 【输出文件】

输出文件 happy.out 只有一个正整数，为不超过总钱数的物品的价格与重要度乘积的总和的最大值 ( $< 100000000$ )。

### 【输入样例】

```
1000 5
800 2
400 5
300 5
400 3
200 2
```

### 【输出样例】

```
3900
```

思路：和采药一样。（如果你不懂采药请看去年的解题报告，或者找一个时间在 10 以内的图表根据程序过程分析算法）但是如果完全照搬在 WINXP 下不能运行，CENA 测的是栈溢出，TC 甚至不让编译通过，说 too big size（好像是系统限制，当时我就是完全照搬的，还好评测时用的是 LINUX，可以运行。）所以要进行优化。回滚，因为这题比较特殊，所以可以用

回滚进行优化，将  $26*30000$  压缩到  $3*30000$  甚至  $2*30000$ （我建议前者，保险）。所谓回滚就是到这一个状态储存在前一个状态的空间里。一个  $\text{mod } 2+1$  就给解决了。这样就避免了空间的浪费。

动态转移方程：i 表示状态，也就是取第几个物品，j 表示总钱数

$$f[i][j] = \max(f[i-1][j - \text{money}[i]] + v[i] * p[i], f[i-1][j])$$

### 3. Jam 的计数法

(count.pas/c/cpp)

#### 【问题描述】

Jam 是个喜欢标新立异的科学怪人。他不使用阿拉伯数字计数，而是使用小写英文字母计数，他觉得这样做，会使世界更加丰富多彩。在他的计数法中，每个数字的位数都是相同的（使用相同个数的字母），英文字母按原先的顺序，排在前面的字母小于排在它后面的字母。我们把这样的“数字”称为 Jam 数字。在 Jam 数字中，每个字母互不相同，而且从左到右是严格递增的。每次，Jam 还指定使用字母的范围，例如，从 2 到 10，表示只能使用 {b, c, d, e, f, g, h, i, j} 这些字母。如果再规定位数为 5，那么，紧接在 Jam 数字“bdfij”之后的数字应该是“bdghi”。（如果我们用 U、V 依次表示 Jam 数字“bdfij”与“bdghi”，则  $U < V$ ，且不存在 Jam 数字 P，使  $U < P < V$ ）。你的任务是：对于从文件读入的一个 Jam 数字，按顺序输出紧接在后面的 5 个 Jam 数字，如果后面没有那么多 Jam 数字，那么有几个就输出几个。

#### 【输入文件】

输入文件 counting.in 有 2 行，第 1 行为 3 个正整数，用一个空格隔开：

s t w

（其中 s 为所使用的最小的字母的序号，t 为所使用的最大的字母的序号。w 为数字的位数，这 3 个数满足： $1 \leq s < t \leq 26$ ， $2 \leq w \leq t - s$ ）

第 2 行为具有 w 个小写字母的字符串，为一个符合要求的 Jam 数字。

所给的数据都是正确的，不必验证。

#### 【输出文件】

输出文件 counting.out 最多为 5 行，为紧接在输入的 Jam 数字后面的 5 个 Jam 数字，如果后面没有那么多 Jam 数字，那么有几个就输出几个。每行只输出一个 Jam 数字，是由 w 个小写字母组成的字符串，不要有多余的空格。

#### 【输入样例】

2 10 5  
bdfij

#### 【输出样例】

bdghi  
bdghj  
bdgij  
bdhij  
befgh

思路：请先看题目：在他的计数法中，每个数字的位数都是相同的（使用相同个数的字母），英文字母按原先的顺序，**排在前面的字母小于排在它后面的字母**。这里是关键，它字母的顺序都是递增的。就是每次都从末位向前查找某一位，这一位与后一位的差大于 1，如果大于

1, 那么这位数字+1 且这位数字后每一位都与前一位构成递增 1 的序列就可以得到下一个 jam 数字。如果连最高位都不能再加了, 那么就没有 jam 数字了。

可能我说得不太清楚。反正思路大体就是这个样子了。还不懂就根据程序推导一下。或者根据联系方式联系我。

## 4. 数列

(sequence.pas/c/cpp)

### 【问题描述】

给定一个正整数  $k$  ( $3 \leq k \leq 15$ ), 把所有  $k$  的方幂及所有有限个互不相等的  $k$  的方幂之和构成一个递增的序列, 例如, 当  $k=3$  时, 这个序列是:

1, 3, 4, 9, 10, 12, 13, ...

(该序列实际上就是:  $3^0, 3^1, 3^0+3^1, 3^2, 3^0+3^2, 3^1+3^2, 3^0+3^1+3^2, \dots$ )

请你求出这个序列的第  $N$  项的值 (用 10 进制数表示)。

例如, 对于  $k=3, N=100$ , 正确答案应该是 981。

### 【输入文件】

输入文件 sequence.in 只有 1 行, 为 2 个正整数, 用一个空格隔开:

k N

( $k, N$  的含义与上述的问题描述一致, 且  $3 \leq k \leq 15, 10 \leq N \leq 1000$ )。

### 【输出文件】

输出文件 sequence.out 为计算结果, 是一个正整数 (在所有的测试数据中, 结果均不超过  $2.1 \times 10^9$ )。(整数前不要有空格和其他符号)。

### 【输入样例】

3 100

### 【输出样例】

981

思路:

从题目可以看出, 第  $n$  个数可以分为两类: 第一类是数  $k$  的一个乘方 (例如  $3^2, 3^3$  等), 第二类是若干个数  $k$  的乘方的和 ( $3^1+3^3$  等)。从中可以找出规律。前者这类数, 第  $n$  个这类数的位置就是 (忽略 0 下标)  $2^{n-1}$ 。即设  $a$  为 1000 个 long 数组元素的数组 (1 开始), 先把所有的第一类数求出来 ( $2^9 < 1000 < 2^{10}$ ), 然后再看第二类数的规律。第二类数是跟在第一类后面的。那么可以总结出, 第  $n$  个第一类数后面的那个数第二类数可以由  $a[n]+a[1]$  得到, 即  $a[n+1]=a[n]+a[1]$ , 以此类推,  $a[n+2]=a[n]+a[2]$ , 一直可以推到  $a[2*n-1]=a[n]+a[n-1]$ , 这样第  $n$  个数之前的数就可以推出来了。这样完全推出 1000 个数, 然后输出要求的那个数就解决了!



计算机这东西很抽象，文字表达不太清楚，如果还有什么疑问就发EMAIL 给我！