

2014 ACM-ICPC Vietnam National Second Round 解题报告

大连市第二十四中学 于纪平

目 录

| | | |
|----------|---------------------------|----------|
| 0 | 概要 | 2 |
| 1 | A - Stock Market | 2 |
| 1.1 | 题目大意 | 2 |
| 1.2 | 算法 | 3 |
| 2 | B - Sum | 3 |
| 2.1 | 题目大意 | 3 |
| 2.2 | 算法 | 3 |
| 3 | D - Treasure Box | 3 |
| 3.1 | 题目大意 | 3 |
| 3.2 | 算法 | 3 |
| 4 | I - Space Tour | 4 |
| 4.1 | 题目大意 | 4 |
| 4.2 | 算法 | 4 |
| 5 | C - ATM Withdrawal | 4 |
| 5.1 | 题目大意 | 4 |
| 5.2 | 算法 | 4 |

| | | |
|-----------|--------------------------------|----------|
| 6 | E - ACM | 5 |
| 6.1 | 题目大意 | 5 |
| 6.2 | 算法 | 5 |
| 7 | H - Pencil Game | 5 |
| 7.1 | 题目大意 | 5 |
| 7.2 | 算法 | 6 |
| 8 | J - Math Magic | 6 |
| 8.1 | 题目大意 | 6 |
| 8.2 | 算法 | 6 |
| 9 | G - Production Planning | 6 |
| 9.1 | 题目大意 | 6 |
| 9.2 | 算法 | 7 |
| 10 | F - Coupled Polygons | 7 |
| 10.1 | 题目大意 | 7 |
| 10.2 | 算法 | 8 |

0 概要

这套题目是Codeforces的一场Gym比赛。

所有的题目可以在<http://codeforces.com/gym/100541>找到。

在这里按照比赛时通过人数降序来分析每道题目。其中前8道题较简单，后2道题较难。

1 A - Stock Market

1.1 题目大意

给出 n 天的股价 p_i 和 W 的资金，要求在某一天买入股票并在另一天卖出，求最大获利。

$$n \leq 100, w \leq 10^6, p_i \leq 1000.$$

1.2 算法

考你会不会编程。枚举买入股票的日期，显然是在之后股价最高的一天卖出。这个可以通过预处理后缀最大值 $O(1)$ 计算。

当然暴力也是可以过的。

2 B - Sum

2.1 题目大意

给出 n ，计算

$$\sum_{i=1}^n \lfloor n/i \rfloor$$

$$n \leq 10^{12}.$$

2.2 算法

显然 $\lfloor n/i \rfloor$ 的取值只会变化 $O(\sqrt{n})$ 次，我们枚举它的取值，计算有多少个 i 取到了这个值就可以了。

3 D - Treasure Box

3.1 题目大意

给出 n 和 k ，问执行 k 次 $n+ = n \bmod 100$ 以后的 n 的值。

$$n, k \leq 10^9.$$

3.2 算法

显然 k 次操作后， n 的增加量只与 $n \bmod 100$ 有关。

令 $f[i][j]$ 表示, 如果 $n \bmod 100 = i$, 那么经过 2^j 次操作后 n 的增加量, 则所有的 f 可以通过倍增算出。

有了 f 我们就可以快速算出 k 次的增加量了。

4 I - Space Tour

4.1 题目大意

给出一个 $n \times m$ 的矩形地图, 有些地方是空格有些是障碍。现在要选一个空的格子作为落地点, 并按照图示规则移动, 遇到障碍则停止。问最多能访问多少个格子。

$$n, m \leq 1000.$$

4.2 算法

我们只需处理出每个点开始向四个方向（左上、右上、左下、右下）能走的最远距离。这个可以十分显然地用动态规划/记忆化搜索得出。

5 C - ATM Withdrawal

5.1 题目大意

有一个纸币系统, 这个系统有一个参数 c , 它的纸币面额是 $1000i * 10^j$, 其中 $i = 1, 2, 3, 5$, $0 \leq j \leq c$ 。

现在要用这些纸币凑出 W 的钱数, 每种纸币充分多, 问最少需要纸币的张数, 及最小前提下的方案数。或输出无解。

$$W \leq 10^{18}, c \leq 15.$$

5.2 算法

无解情况就是 $W \bmod 1000 \neq 0$ 。对于有解情况我们先把 W 和所有面额都除以 1000。

先考虑 c 充分大的情况。此时可以证明，最优策略一定是把 W 按照十进制拆分成若干个 $i * 10^j$ 的和，然后分别用若干张 $k * 10^j$ 的纸币去凑这个 $i * 10^j$ 。（也就是说，保证全过程不会发生任何进位）这个可以通过代入 $W = 11, 12, \dots$ 等较小的数验证一下。

如果 c 比较小，我们最后就会遇到一个问题是，用1, 2, 3, 5凑出 W ，其中 W 可能很大并且至少为10。显然最优策略是用很多个5，那么只需根据 $W \bmod 5$ 的值分类讨论求出最优策略即可。

6 E - ACM

6.1 题目大意

给出一个长度为 n 的序列，一开始都是1，每次可以把一个区间的数乘上某个 x ，或除以某个 y （保证能整除），或询问一个区间的积对 p 取模的答案。不同询问的 x, y, p 不一定相同。

$$n, m \leq 50000, 1 \leq x, y \leq 150, 1 \leq p \leq 10^9 + 7。$$

6.2 算法

注意到修改操作的 x, y 都不会很大，我们对150以内的每一个素数都维护一个序列，记录这个位置有这个素数的多少次幂的因子。那么三种操作都可以转化为序列的区间增减和区间求和问题，可以用线段树解决。

7 H - Pencil Game

7.1 题目大意

$n \times m$ 的矩阵（下标从0开始），第 i 行第 j 列的元素是 $i * m + j$ 。现在给出 L ，要找出面积最小的子矩形使得元素和为 L ，或输出无解。

$$n, m \leq 10^6, L \leq 10^{12}。$$

7.2 算法

注意到 $2L$ 显然是所求子矩形面积的倍数，那么我们在 $2L$ 的因数当中从小到大枚举所求的面积。

枚举出面积以后，再从它的因数中枚举子矩形的横向长度，从而确定它的形状。容易得出这个子矩形的元素和与右下角元素是线性关系，那么我们解这个一元一次方程并判断解的合法性，就能知道是否确实存在一个这样形状的矩形。

对于枚举出的所有情况取最优解就可以了。

8 J - Math Magic

8.1 题目大意

有一个十字形状的东西，它的四个端点每个都有一种颜色（红黄蓝绿之一）。现在有 n 个东西要依次往上接，每个东西也是十字形，端点也有颜色。要求只能接在 x 轴或 y 轴上，且与原来的东西的接点必须是相同颜色的。每次接上，会根据公共接点的颜色获得某个权值；也可以选择不接，获得另一个权值。

问获得的最大权值之和。

$n \leq 250000$ 。

8.2 算法

我们用五元组 $(i, j_{x+}, j_{x-}, j_{y+}, j_{y-})$ 来表示一个状态， i 表示现在放的物品编号， j_{x+} 表示 x 轴正半轴的最靠上的接点颜色，其余同理。

根据这个状态来做动态规划就可以了。

9 G - Production Planning

9.1 题目大意

有 $n-1$ 种原材料和 n 种产品，生产1个产品 j 需要用到原料 i 的个数为 $a_{i,j}$ ，可以卖 c_j 的钱。

现在第 i 种原料有 b_i 份，求花光所有原料的前提下最大的获利。注意只能生产整数份产品。

$n \leq 200, c_j \leq 1000, a_{i,j}, b_i < 10^6$ ，保证 a 的秩是 $n - 1$ 。

9.2 算法

假设产品 n 的产量是0，那么根据题目条件，我们可以用高斯消元算出用光所有原料的一种唯一方案。但是这个方案中某些产品的产量可能是小数，甚至是负数。

再假设产品 n 的产量是1，我们也可以算出这样的一种“方案”。

我们可以得出，产品 n 每多1件时，其他每种产品产量的变化量。

枚举产品 n 的产量，我们就可以算出其他产品的产量。如果都是非负整数那么就找到了一个合法方案。在合法方案中取最优就可以了。

但是这样会超时，因为枚举的产量可能是 10^6 级别的。我们只需选一个最大可能产量最小的产品枚举就可以了。

我们要面对的问题是，如何判断一个浮点数是不是整数？很不幸double和long double的精度都挂掉了。

我们选两个模数，分别在这两个模意义下跑以上的算法，如果算出的答案一样就很可能是整数，不一样就一定不是整数。

在模数较大时这个算法的正确率是很高的，可以安全通过。

10 F - Coupled Polygons

10.1 题目大意

一个每条边都与坐标轴平行的简单多边形，可以用一个由NESW组成的字符串表示。其中每个字符表示按照这个方向画一条长度为1的线段。

现在给出这样的多边形 A ，求一个多边形 B ，使得：

- A 和 B 可以通过平移，来完美地拼成一个正方形；
- 在此基础上，拼成的正方形的边长应该尽可能小；
- 在此基础上， B 的字典序应该尽可能小。

输入串的长度不超过100000。

10.2 算法

设 A 的较长边长度为 a ，较短边长度为 b ，那么答案正方形的边长一定是 a ， $a+1$ 或 $a+2$ 。我们从这三者中依次枚举答案。

枚举正方形边长后，考虑正方形相对于 A 的位置。在较长边 a 的方向上显然只有 $O(1)$ 个位置，而在较短边 b 上可能会有很多位置。不过有意义的位置只有贴边界和离边界一格，因为其它的位置合法性不会改变，而且也不可能字典序最小。由于有两个边界，所以一共有4个位置，还是 $O(1)$ 个。我们依次枚举这些 $O(1)$ 个位置。

枚举出正方形的大小和位置后，我们要判断 A 在正方形中的补是否是多边形。

我们考虑 A 和正方形的边的交。如果交为空则显然补不是多边形（中间有一个 A 形状的洞）。如果交不为空，那么， B 是多边形等价于交是 A 中的连续若干条边。这个容易判定的。

判定出合法性以后，我们要选出字典序最小的多边形的字符串表示。这只需要任取一种表示后算出它的循环串最小表示。这是经典问题，可以用后缀数组等方法解决。

对于这 $O(1)$ 个枚举出的位置取最优解即可。