

Bestcoder Round31 解题报告

浙江省余姚中学 张浩威

1 摘要

这是我在bestcoder上出的第二套题目，在这里分享一下出题的过程，试题可以在http://bestcoder.hdu.edu.cn/contests/contest_show.php?cid=569找到。

2 1001

2.1 题目大意

在X轴上有 n 个点，他们的坐标分别为 $x[i]$ 。求存在多少点对 (a,b) 满足 $|x[b] - x[a]| \leq k (a < b)$ 。共有 T 组数据。

2.2 数据范围

$$1 \leq T \leq 5, 1 \leq n \leq 100000, 1 \leq k \leq 10^9, -10^9 \leq x[i] \leq 10^9.$$

2.3 解析

一个比较暴力的做法是枚举所有点对，并进行统计满足条件的对数，但这种做法时间复杂度是 $O(n^2)$ 的，难以通过本题。

如果我们将 x 先从小到大进行排序，排序完之后注意到 $x[b]-x[a]$ 一定是非负整数。 b 不变时，满足条件的 a 一定是连续的，且随着 b 向右不断增加， a 的左端点也在向右移动。

因此我们用two point在排序完之后线性求得答案。

时间复杂度为排序的复杂度即 $O(n \log_2 n)$ 。

与上场的1001比较，难度提升了很多，主要是考虑到后面2题比较难，增加题目区分度。

3 1002

3.1 题目大意

令 $A = \sum_{i=1}^n a_i * 10^{n-i}$ ($1 \leq a_i \leq 9$) (n 为 A 的位数)。若 A 为“漂亮的数”当且仅当对于任意 $1 \leq i < n$ 满足 $a[i] \geq a[i+1]$ 且对于任意 $1 \leq i \leq n, i < j \leq n$, 满足 $a[i] \bmod a[j] = 0$ (例如 931 是一个“漂亮的数”而 87 不是), 求在区间 $[L, R]$ (包含 L 和 R) 中“漂亮的数”的个数。共有 T 组数据。

3.2 数据范围

$$1 \leq L \leq R \leq 10^9, 1 \leq T \leq 100$$

3.3 解析

我们发现题目给定的两个条件实际上是非常强的。

如果我们将所有符合条件的解都打印出来, 可以观察到只有 1299 个。

因此我们可以打一个表, 每次查询时只要在这张表中找满足条件的解的个数就可以了。

另外也有数位 dp 的方法。具体地, 可以令 $dp[i][k]$ 表示从低到高第 i 位, k 是一个二进制状态, 表示当前这个数中 1 至 9 是否已经出现过, 转移比较简单。

但是数位 dp 的做法代码十分繁琐, 远远没有暴力打表来的方便。

该题在设计时主要考察的是选手的应变能力, 因为 Bestcoder 是按时间与错误次数计分的, 如果选择数位 dp 去做这个题目, 不仅难写也很容易写错。如果换一个角度思考, 就会发现更优秀方便的解法。

4 1003

4.1 题目大意

在一个 $n*n$ 的国际象棋棋盘中, 放置若干国王与 k 辆车, 使得国王之间不相互攻击, 车攻击不到国王 (但国王是能攻击车的)。国王能攻击它八联通的位置, 车能攻击它同一列同一行的位置。求方案总数对大质数取模。共有 T 组数据。

4.2 数据范围

$$1 \leq T \leq 10, 1 \leq n \leq 15, 0 \leq k \leq 15.$$

4.3 解析

解决该问题的关键在于“国王能攻击到车”，因此摆放一辆车相当于是将棋盘划分成4小块。

假如我们预处理出所有 $x*y$ 的格子放若干国王的方案总数，那么将整个棋盘划分成若干块，根据乘法原理将每一块的方案总数都乘起来就是方案总数了。

令 $dp1[i][j][k]$ 表示有 i 行 j 列的格子，当前第 i 行国王的放置状态为 k 。转移只需枚举下一行放置国王的的状态就可以了。

如果我们是这样暴力转移，时间复杂度为 $O(n * m * 2^{2n})$ ，无法通过本题，但注意到国王是不能相邻的，因此 k 的状态数最多只有斐波那契第15项这么多，足以跑出解。

接下来我们令 $dp2[i][j][k]$ 表示一个 i 行 j 列的矩阵，但是 k 这个状态的行不能放置国王的方案总数，这是可以线性扫出来的。

这样我们枚举哪些行有车，哪些列有车，通过 $dp2$ 数组来不断计算方案，时间复杂度为 $C(15, 7)^2$ ，但有大量乘法操作与取模操作，无法通过本题，不过由于题目中只需读入两个数，并且范围都不是很大，直接打表就行了。

当然也有不需要打表的做法，当我们枚举的行确定后，是可以通过 dp 来解决的，令 $dp3[i][j]$ 表示当前在第 i 列，且已经选择 j 列放车的方案总数(第 i 列一定放车)，之后枚举前面放车的列转移就可以了。时间复杂度为 $C(15, 7) * n^2k$ 。

本题是一道思维难度较高的题，需要选手不断挖掘题目性质，在考场中仅有一名选手是不打表通过全部解的。

5 1004

5.1 题目大意

有 n 个数按顺序依次进栈，且在某个时刻从栈顶弹出。

有 m 个限制，每个限制形如 (A, B) 表示第 A 个数必须在第 B 个数之前出栈。

求方案总数，答案对大质数取模。

共有T组数据。

5.2 数据范围

$$1 \leq n \leq 300, 1 \leq m \leq 90000, 1 \leq A, B \leq n, 1 \leq T \leq 5.$$

5.3 解析

我们先考虑 $m=0$ 的情况。

令 $dp[i][j]$ 表示仅有第 i 个数至第 j 个数依次进栈的方案总数。

枚举 k 表示最后出栈的元素，有 $dp[i][j] = \sum_{k=i}^{j-1} dp[i][k] * dp[k+1][j]$ ，实际上就是卡特兰数列，时间复杂度为 $O(n^3)$ 。

考虑所有限制条件的本质，若 $A < B$ ，意味着 A 不能作为最后一个出栈的，若 $A > B$ ，意味着 $B+1$ 至 A 不能作为最后一个出栈的（这里只针对 $i \leq \min(A, B), j \geq \max(A, B)$ 而言）。

这样我们就得到了一个 $O(n^3m)$ 的做法。

考虑优化这一个做法。

令 $v[i][j][k]$ 表示进行 $dp[i][j]$ 的转移时， k 能否作为最后一个出栈的元素。注意到对于所有限制，不能作为最后一个出栈的元素一定是连续的，因此可以枚举 i 与 j ，通过并查集来加速 k ，时间复杂度降为 $O(n^2m)$ ，但仍然通过不了本题。

继续挖掘性质，实际上不仅 k 是连续的， i 与 j 也是连续的。因此这本质上构成了一个空间。我们只需在数组中将这个空间描述出来就可以了。

描述的方法可以通过记录所有端点，最后进行前缀和统计。

计算完 $v[i][j][k]$ 后，就可以直接进行 dp 啦。

时间复杂度为 $O(n^3)$ 。

本题与上题类似，需要选手挖掘题目性质，不断进行优化。

6 得分情况

共133人通过1001,123人通过1002,5人通过1003,1人通过1004。

7 总结

这套题目的标程都不长，考察的都是选手的思维能力，得分情况与预估相似。

与上套题目相比，更好的区分了高层次选手之间的水平。