

山东大学2016级新生ACM程序设计竞赛

命题的恐怖分子



Solution

December 3rd, 2016

1 Problem A: 9 Pound 15 Pennies

Abstract Description

在描述区给了你一篇文章，要求你输出其中包含字母H或h的单词的个数。

Result

解题情况：362次提交，74人通过。

Solution

答案是25。

我们在评测过程中，发现不少选手对题意产生了误解，比如有的选手读入了一个字符串并与"None."作比较，相等再输出25。其实这都是经验（套路）不足，还需要学习一个的表现。

2 Problem B: Patchouli and Sakuya

Abstract Description

有 N 块每块能保存 M 个二进制位的存储器，要求维护以下5种操作：

- 将某块的某二进制位置为1
- 将某块的某二进制位置为0
- 反转某块存储器上所有的二进制位
- 交换两块存储器中的内容
- 将所有存储器返回到某个操作之后的状态

并在每次操作后输出所有存储器中值为1的二进制位的个数。

Result

解题情况：11次提交，无人通过。

Solution

先介绍在线和离线的概念：

通常对于有多组询问/操作的题目，解法会有「在线」和「离线」的区别。这里的「在线」和「离线」与QQ中的同义词含义全然不同：

- 在线：每读入一组询问或操作后立即处理之。
- 离线：读入所有询问或操作后，将它们以某种方式重新组织起来，再按某种特定顺序处理。

此题的在线做法为可持久化线段树。然而秉持着「新生赛不能考具体算法和数据结构」的思路，我们来考虑一种离线做法。在此做法中，我们把每次操作完成后所有存储器的状态记为树上的一个节点，并将初始状态视为根节点。在添加操作时，我们将新节点附于表示它上一个状态的节点之下，即：对于前四个操作，直接将新节点作为上一个操作结束后状态的子节点；对最后一种操作，让新节点作为所返回的操作结束后状态的子节点。并将对应的操作保存在边上。参见下面这个例子：



（第一行为 N, M, Q ，下面9行为9个操作。右图中节点上的两个字母分别代表两个存储器中二进制位的状态：O代表1，X代表0。）

于是将所有操作建成这种类似树的结构之后，我们就可以从根出发进行一次dfs（深度优先搜索），沿边下行时执行边上的操作，回溯时执行边上操作的逆操作（即撤销此操作）。这样我们就可以按照这棵操作树的dfs序的顺序得出每步操作后的结果。将结果保存在一个数组中，最后按顺序输出即可。

由于dfs的复杂度本身为 $O(Q)$ ，所以对于反转、交换以及统计结果这些操作需要做到 $O(1)$ 处理。于是我们可以记录翻转标志 I_a 和位置标记 B_a 。这样第 a 个存储器中第 b 个二进制位中实际表示的值即为 $b_{B_a, b} \oplus I_{B_a}$ （ \oplus 表示异或操作，初始时显然有 $B_i = i$ ）。这样我们在反转操作时只需将相应的 I_{B_a} 取反，交换时只需交换 B_{a_1} 和 B_{a_2} 即可。而由于所有存储器的空间相同，故反转操作后的计数可以通过记录每个存储器中当前1的个数来完成。

到这里我们已经基本完成了这道题目，然而还有一点需要注意的地方：无效操作。注意到上面的例子中，第五个操作在执行时并没有造成任何变化。然而如果在回溯时直接进行原操作的逆操作就会导致错误。所以在操作执行时如果遇到了无效操作，则需要特殊处理。

总体复杂度为 $O(Q)$ 。

3 Problem C: Polygon

Abstract Description

给出3个二维平面上的点 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ ，问这3个点是否有可能取自一个正 N 边形的顶点集。只需考虑 $N \in [3, 5]$ 的情况。

Result

解题情况：82次提交，2人通过。

Solution

方法一：分类讨论。

若三个点之间距离相等，则为正三角形。若存在两边长度相等且其长度平方之和等于第三边长度的平方（或其夹角为90度），则为正四边形。若存在两边长度相等且其夹角为108或36度，则为正五边形。其他情况则直接输出-1。

方法二：????

显然，若给定的三个点取自一个正多边形，则这三个点组成的三角形的外接圆也是原正多边形的外接圆。故可以求出给定三点组成的三角形的外接圆圆心，然后计算任意两点和圆心组成的夹角大小是否为120度/90度/72度的整数倍即可。此方法可以推广到 $N > 5$ 的情况。

由于题目中已经提到的原因，不论采用何种方法，此题（乃至大部分使用浮点数类型变量的题目）在所有判定实数相等时均需要注意浮点数误差。

4 Problem D: Weight

Abstract Description

给定取自一个按 $30, 29, 28, \dots, 3, 2, 1, 2, 3, \dots, 28, 29$ 规律循环的序列，判断给定序列的下一个数相比给定序列的最后一个数会增加还是会减少。

Result

解题情况：371次提交，60人通过。

Solution

给定序列的最后一个数为1→增加；
给定序列的最后一个数为30→减少；
给定序列长度为1且不满足以上两条→不确定；
其他情况→与序列的最后一个数和倒数第二个数的关系相同。
注意第三条中的「不满足以上两条」是一个坑。

5 Problem E: Gua and Hui

Abstract Description

一张被划为 $W * H$ 网格的纸，两个玩家轮流选择一个位置将一片纸沿网格的边缘撕开。当轮到其中一方操作时，如果没有再能撕的纸（即所有纸均为 $1*1$ ）则输。问在两方均采用最佳策略的情况下，谁会取胜。

Result

解题情况：102次提交，34人通过。

Solution

其实它根本不是博弈！

定理1:

按照题目中的规则，将一张纸全部撕成 $1*1$ 总是需要 $W * H - 1$ 步。

所以当 $W * H$ 为奇数时后手胜，否则先手胜。

6 Problem F: A hard problem

Time Limit: 1000ms

Abstract Description

给出 N 对数 $a_1, b_1, a_2, b_2, \dots, a_N, b_N$ ，请按每对数的和升序给这些数对排序。和相等者则维持输入时的相对顺序。

Result

解题情况：189次提交，13人通过。

Solution

方法一：稳定排序

排序算法的稳定性表示对于等价元素，该排序算法进行完毕后它们的相对位置是否会发生变化。

一般来说，冒泡排序、插入排序、归并排序以及C++中的`std::stable_sort`和Java中为对象数组排序的`sort`方法是稳定的，而选择排序、快速排序以及`stdlib.h`中的`qsort`，C++中的`std::sort`是不稳定的。

由于题目中 $1 \leq N \leq 1000$ ，所以可以任选一种稳定的排序来解决此题。

方法二：多关键字排序

将两数的和作为第一关键字，将数对在原序列中的位置作为第二关键字，进行一次多关键字排序即可。这种方法显然和排序的稳定性无关。

注意套路： $a_i, b_i < 10$ ，但是数据可以用64位有符号整数存下。

7 Problem G: Hui's GPA

Abstract Description

给你一个正实数 N ，现允许将其四舍五入至小数点任意位（包括四舍五入到整数） K 次，问在至多 K 次操作之后，结果的最大值是多少。

Result

解题情况：25次提交，一人通过。

Solution

容易发现，只有形如 $\overline{444\dots 444a}$, $a \geq 5$ 这样的一段才会需要多次连续进位。所以我们只需要找到位于小数点后且最靠近小数点、长度小于等于 K 的符合上述形式的串，从它的最后一位开始进位即可。注意输出的结果需要符合书写习惯：两端不要有多余的0，小数点不要作为最后一个字符等。设 $L(N)$ 表示 N 的长度，则总体复杂度为 $O(L(N))$ 。

8 Problem H: A boring game

Abstract Description

一个游戏的规则如下：有两个数字 A 和 K 分别代表当前分数和当前等级。初始时 $A = 2, K = 1$ 。有两个按键，按下第一个按键的效果是 $A += K$ ，按下第二个按键的效果是 $A = \sqrt{A}, K += 1$ （仅在 A 为一完全平方数时可用）。在每次按键之后，需要保证 $K \mid A$ 。求想要达到 K_t 级的按键方案。

Result

解题情况：5次提交，一人通过。

Solution

考虑一种构造答案的方法。由题意发现在按第一个键的过程中，容易保证 $K \mid A$ 。所以只需保证在按下第二个键时 A 为完全平方数且 A 含有两个 $K + 1$ 因子。又因为 A 总为 K 的倍数，不妨使在按下第二个键前 $A = (K(K + 1))^2$ 。这样在按下第二个键后 $A = K(K + 1)$ ，满足题意。按此方法构造答案，从 K 级升至 $K + 1$ 级（ $K > 1$ ）需要按第一个键 $K * (K + 1)^2 - (K - 1)$ 次。 $K = 1$ 时特殊处理即可。时间复杂度 $O(K_t)$ 。

9 Problem I: niconiconi

Abstract Description

假设一个字符串的第奇数个字母与它后面的一个字母组成一个音节。给定一个长度为偶数的字符串，请从音节角度判断这个字符串是否是回文的。

Result

解题情况：136次提交，55人通过。

Solution

假设给定的字符串长度为 L ，则对所有 $i \in [0, L/4)$ ，检查是否有 $s[2*i] == s[L-2*i-2]$ 和 $s[2*i+1] == s[L-2*i-1]$ 即可。亦可以将每两个字符以某种方式组合起来并视作一个数字（要求每个数字只能对应一种字母组合），并直接判断所得的数字串是否为回文。

10 Problem J: Walloping the problem setters

Abstract Description

有 N 条线，求 N 次随机取两个线头接在一起后，恰好将所有线系成一个圈的概率。

Result

解题情况：61次提交，17人通过。

Solution

容易发现对该问题中，对 $N = n$ 时问题的答案可以由 $N = n - 1$ 的答案得到：只要第一次操作并非直接将一根线接成环，则直接能变为等价于 $N = n - 1$ 的情况。设 $N = n$ 时的答案为 $f(n)$ ，则 $f(n) = f(n - 1) * \frac{2 * (n - 1)}{2 * n - 1}$ 。显然 $f(1) = 1$ 。于是直接暴力求解该递推式的解即可。时间复杂度 $O(N)$ 。

