

挑战 (challenge)

【问题描述】

你和同学们找了三道题目用来练习。

这次练习的目标是写出能在时间限制里通过尽量大规模数据的代码。

同学们纷纷写出了优秀的代码。现在，他们向你发起了挑战，他们对每个问题都设置了若干个测试数据，这是他们能通过的最大规模的测试数据。现在，他们想看一看你写的代码究竟能超过多少同学的代码，通过多大规模的测试数据。

本题分为3个任务，每个任务对应一道题和相应的若干个测试点，你需要对于每个任务，设计一个能通过尽量多测试点的程序。

任务1

给定 n 个32 位无符号整数，将它们从小到大排序。

任务2

有 $2n$ 个人在玩“石头剪刀布”游戏。他们排成两排，每排 n 个人。每个人在每一局游戏都使用固定策略，即对于第 $i(i \in \{1, 2\})$ 排的第 $j(0 \leq j < n)$ 个人，用一个整数 a_{ij} 表示他的策略，其中0 表示只出石头，1 表示只出剪刀，2 表示只出布。

现在有 q 个询问，每个询问给定三个整数 $x, y, l(0 \leq x, y < n, 1 \leq l \leq n - \max(x, y))$ ，问将第一排的第 $x \sim x + l - 1$ 个人和第二排的第 $y \sim y + l - 1$ 个人比赛之后，第一排有多少个人会赢。

上文中“比赛”的意思是，对于所有整数 i 满足 $0 \leq i < l$ ，让第一排的第 $x + i$ 个人和第二排的第 $y + i$ 个人进行“石头剪刀布”游戏。

任务3

我们称一个合法的括号串为：只由左括号和右括号构成，两种括号的数量相等，且任意一个前缀的左括号数量不少于右括号数量的串。现在给定一个由“(”、“)”和“?”构成的串，问有多少种不同的方案，使得将每个“?”都替换成一个括号之后，该串变成一个合法的括号串。两种方案不同，当且仅当至少有一个位置的“?”被替换成了不同的括号。

由于本题部分任务数据规模比较大，我们用一些特殊的方式进行输入输出。先定义一些将要用于输入/输出的函数（以下代码中除法向下取整，数组下标从0开始，异或为按位异或，右移为逻辑右移）。

定义函数 `next_integer(32位无符号整数 x)`：

将 x 赋值为 x 异或 $(x$ 左移 13位)

将 x 赋值为 x 异或 (x 右移 17位)
 将 x 赋值为 x 异或 (x 左移 5位)
 返回 x

定义函数 `output_arr`(指针 a , 32位无符号整数 $size$):

如果 ($size$ 模 4) 不等于 0:

输出 -1

返回 假

令 32位无符号整数 $count$ 等于 $size$ 除以 4

令 32位无符号整数指针 b 等于 a 强制转换为 32位无符号整数指针

令 32位无符号整数 ret 等于 $size$

令 32位无符号整数 x 等于 23333333

令 32位无符号整数 i 从 0 循环至 $count - 1$:

将 ret 赋值为 ret 异或 ($b[i]$ 加 x)

将 x 赋值为 `next_integer(x)`

输出 ret

返回 真

【输入格式】

从文件 **challenge.in** 中读入数据。

第一行一个整数 $task_id$ ($1 \leq task_id \leq 3$), 表示任务编号。接下来是每个具体任务的输入内容。

在输入的同一行中, 相邻的两个整数会被一个空格隔开。

对于任务1: 一行, 两个整数 n, s 。令 $a_0 = \text{next_integer}(s)$, $a_i = \text{next_integer}(a_{i-1})$, $1 \leq i < n$, 则 a_0, a_1, \dots, a_{n-1} 即为需要排序的 n 个整数。

对于任务2: 第一行两个整数 n, q 。第二行一个仅包含“0”, “1”, “2” 的长度为 n 的字符串, 第 i 个字符所代表的整数表示第一排第 i 个人的策略 (即 a_{1i})。第三行格式同第二行, 表示第二排各个人的策略。

对于任务3: 第一行一个整数 n , 表示给定的串的长度。第二行一个字符串, 即为给定的串。

【输出格式】

输出到文件 **challenge.out** 中。

对于任务1: 令 b_0, b_1, \dots, b_{n-1} 为排好序的数组, 调用 `output_arr(b, n * 4)` 即可。

对于任务2: 将每个询问的答案依次存入 **32位无符号整数** 数组 b 中 (即, 存入 b_0, b_1, \dots, b_{q-1} 中), 然后调用 `output_arr(b, q * 4)` 即可。

对于任务3： 输出一个整数，表示不同的方案数除以 2^{32} 得到的余数。

【数据范围和约定】

本题共有9个测试点，不同测试点的分值可能不同。对于每个测试点，只有当你的输出与标准输出完全相同时，才能得到该测试点的满分。下表为各个测试点的数据范围和约定。为了方便阅读，“测试点编号”一列被放到了表格的中间而不是左边。

任务编号	分值	测试点编号	数据范围和约定
1	5	1	$n = 100000$
	19	2	$n = 10^8$
	11	3	$n = 2 \times 10^8$
2	7	4	$n = q = 1000$
	23	5	$n = q = 300000$
3	9	6	$n = 1000$
	5	7	$n = 120000$
	7	8	$n = 225000$
	14	9	$n = 266666$

【提示和说明】

对于所有语言都打开-O2 编译开关。

所有测试点的时间限制均为3s。所有测试点的空间限制均为2GB（对于32位计算机，该限制即为单个程序能使用的内存的上界）。

评测在本机进行。请注意，所有机器的配置与型号都是相同的。

对于每种编程语言，我们都提供了一个只实现了输入输出功能的模板程序，名为challenge_model.pas/c/cpp。你可以在这个模板程序的基础上开始解题，也可以不使用这个模板程序，这与你的得分无关。

我们提供了一些样例输入输出文件，名为challenge1.in/ans ~ challenge9.in/ans，其中第*i*组样例输入的数据范围与第*i*个测试点的数据范围相同。各测试点的数据范围详见【数据范围与约定】。

请注意，根据NOI系列比赛的有关规定，你提交的代码长度不能超过100KB。

【任务详细说明】

我们在这里对一些任务中的一些可能引起疑问的地方进行了较为详细的说明。

对于任务1：

32位无符号整数即占用内存空间32个二进制位（或4字节）的无符号整数类型，在C/C++中为unsigned int，在Pascal中为dword。

从小到大排序是指，将 n 个整数 a_0, a_1, \dots, a_{n-1} 重新排列，设排列后这些整数为 b_0, b_1, \dots, b_{n-1} ，则满足对于任意 $1 \leq i < n$ ，都有 $b_{i-1} \leq b_i$ 。

对于任务2：

“石头剪刀布”游戏的规则是石头赢剪刀，剪刀赢布，布赢石头，除这三种情况以外都不算赢。

输入数据和询问中，每一排人的下标都是从0开始的，即“第 $x(0 \leq x < n)$ 个人”表示的是这一排从左到右数的第 $x + 1$ 个人。

对于任务3：

我们给出一些括号串的例子：“(())(())”是合法的，“((((())()))”也是合法的；“(((“是不合法的，“()”(也是不合法的。

对于输入数据“(???”，答案为2，有“(())”和“()()”两种方案。对于输入数据“)???”，答案为0，因为不可能有合法的括号串以右括号开头。