# RUC ICPC 2020 Individual Contest Solution（标准组+ 硬核组）

RUC 2020校赛命题委员会

2020 年 12 月 25 日

## Contents

# A Ogi的ID

## A.1 Description

给你一个字符串，要你判断它是否满足：

- 以若干个O或o开头；
- 接着是若干个g；
- 然后是若干个i结尾.

## A.2 Solution

简单题，从左至右，模拟自动机转移即可。

# B 调分

## B.1 Description

n个数$\{a_n\}$，每次可以令某一个$a_i$增加1或减少1，求最少的操作次数，使得所有$a_i$的中位数恰好等于s(给定).

## B.2 Solution

先对数组进行排序，然后考虑s在排序后的数组中的位置。排除s比所有数都小/ 比所有数都大的情况（此时挑最近的一半的数变成s即可），假设$a_k \leq s \leq a_{k+1}$，如果k 在n 的左半边，则挑$a_{k+1}..., a_{n/2+1}$ 变成s，这样小于等于s 的数就有一半了，而且由于排过序，这样做的代价一定最小；k 在n 的右半边的情况也类似。

# C 合并数字

## C.1 Description

给定$n$个数字组成的序列$a_i$，有A和B两个玩家，每个玩家每次将序列左边$m$个数字删除，并将这$m$个数字之和加入序列最左边，自己获得这$m$个数字之和作为分数。两个玩家轮流进行，A先手，问A最多领先B多少分。

### C.2 Solution

这是一道动态规划题目，用$f[i][j]$表示取完前$i$个数字且最后一次是玩家$j$操作的情况下，玩家A最多领先玩家B多少分。其中$j = 0, 1$，0代表玩家A，1代表玩家B。

转移方程如下：

$$f[i][0] = \max(f[i-1][0] + a[i], f[i-1][1] + sum[i]),$$

$$f[i][1] = \max(f[i-1][1] - a[i], f[i-1][0] - sum[i]),$$

其中$sum[i] = \sum_{j=1}^{i} a[j]$。以$f[i][0]$的转移方程为例，玩家A可以把第$i$个数并入之前的自己合并，也可以在玩家B合并完前$i-1$个数之后合并。$f[i][1]$的转移同理，改为减号是因为玩家A会损失那么多领先。

## D 地毯

### D.1 Description

一个$2^n * 2^n$ 的表格，其中每个数生成满足一定规律（详见题面），询问若干次表格中一块矩形区域中数字的异或和。

### D.2 Solution

首先，可以发现生成规则中“复制左上角的矩阵再加上同一个数”实质上是把原本0..$2^{2k}-1$的最高位后面两位00进行扩展（01，10，11），扩展后的数正好对应0..$2^{2k+2}-1$.

因此，已知坐标(x,y)，我们可以通过逆向推导生成规则得到对应的数：已知现在处理$2^k * 2^k$的区域，我们判断一下(x,y)属于哪个矩阵，相应地就能得到扩展位的值了，再不断向下逆推$2^{k-1} * 2^{k-1}$, ...，$2 * 2$.

实际上，如果记x的二进制表示$a_k a_{k-1} ... a_1 a_0$，y的二进制表示$b_k b_{k-1} ... b_1 b_0$，则(x,y)值的二进制表示为$a_k b_k ... a_1 b_1 a_0 b_0$，这说明x和y对值的贡献是独立的。因此，对于一行数（$x1, y1$)到$(x1, y2)$ 的异或和，假定总共有偶数个数（奇数类似），则所有x的部分出现偶数次全抵消掉，变为求$y1...y2$的异或和，这个求解是o(log y) 的。

一个更强的结论：所有长、宽均为偶数的矩阵的异或和等于0。证明与上面类似，先按行算，所有x的贡献会抵消掉，再按列算，所有y的贡献也会抵消掉。所以对任意一个矩阵，我们总可以拆成一个偶数长宽的矩阵加上1行或1列的形式，只需计算1行或1列即可。

求1..A的异或和S：求A的二进制形式，如果末位为0，则S除末位的其他位不变；如果末位为1，则其他位为0；S末位的值等于A第一位和第二位的异或和。

# E 比赛

## E.1 Description

给你一个数列w，初始你在坐标0，假设第i-1天的坐标是$f_{i-1}$，则第i 天的坐标为$f_i = \max(f_{i-1} + w_i, i * K)$. 现在要你支持两个操作：修改某个$w_i$的值，或输出某个$f_i$的值。

## E.2 Solution

假设现在要求第t天的里程，令$h_i = (i-1) * K + \Sigma_{j=i}^{t} w_j$，则第t 天的里程实际上等于$\max_{i \le t+1} h_i$.

我们用线段树维护序列h, $h_i = (i-1) * K + \Sigma_{j=i}^{n} w_j$，再用一个树状数组维护w数组，对于询问t，我们计算区间最值，再减去w[t+1..n]的后缀和；对于修改，对h和w相应做区间修改。

# F 联机

## F.1 Description

$1 \sim n$上给你若干个区间，要你统计有多少数对所属的区间的并有交。

## F.2 Solution

将区间以左端点为第一关键字，右端点为第二关键字排序，注意到若有区间被其他区间包含，那么这个区间是无用的，所以我们保证: $l_i \le l_{i+1}, r_i \le r_{i+1}$。

从左到右扫一遍，每次可以统计$l_i \sim l_{i+1}$这些点最远可以到$r_{i+1}$, 有$O(1)$个细节

# G 逃离地牢

## G.1 Description

给定一个矩阵，求出包含左上角与右下角联通块，使得联通块内的元素的$gcd$最大，并且求出方案数。

## G.2 Solution

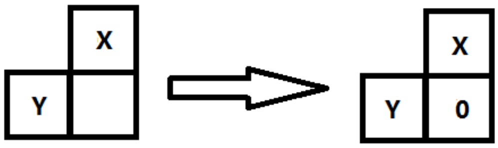
看到有一维很小，即考虑状态压缩，再加上要求维护联通性，即发现是插头Dp裸题。

由于值域只有1000,预处理所有的$gcd$。然后考虑宽最多只有6，直接最小表示法将轮廓线上的联通状态压缩起来，记为状态S。那么我们设计状态$f_{i,j,S}$,表示推到了第$i$行第$j$个格子，当前

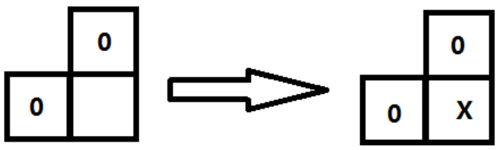
扫描线的联通状态为S，f记录当前状态的$gcd$与数量。我们记与左上角联通的状态为1。

那么转移也就很简单了，假设我们推到了第i行第j个格子，通过S我们可以得到该格子上方与左方的格子的联通性，于是我们做如下讨论：

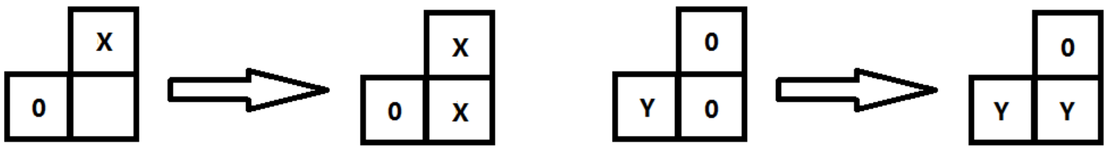
1.只要此处不是最左上角or最右下角的格子，就可以不走，那么我们将S第j位的状态变为0即不连通。



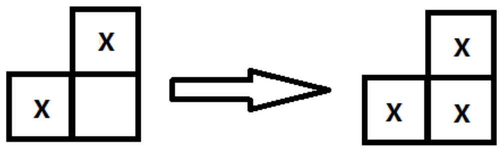

2.若上方与左方的格子均不连通，则在此处新建一个联通状态，即将S第j位的状态变为轮廓线上未出现过的一种联通状态(最小表示法)。



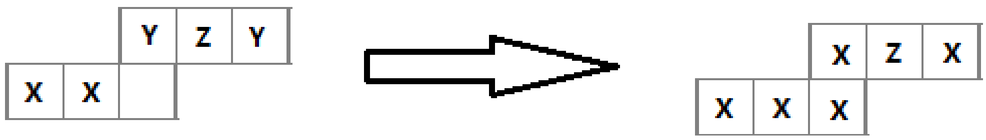

3.若上方与左方的格子仅有一个连通，则将此格的状态修改为与之相同的状态，即将S第j位的状态变为与连通格子相同的状态。




4.若上方与左方格子均联通且连通性相同，则将此格的状态修改为与它们相同的状态，即将S第j位的状态变为与上方和左方相同的状态。




5.若上方与左方格子均连通且连通性不同，则此格取数字较小的状态，同时将轮廓线上与上方和左方相同的状态都变为此格的状态。




# H Masou象棋

## H.1 Description

一个5*5的棋盘，有一个空格，每次走子可以用空格8个方向（左2上1、左1上2、右1上2、右2上1、右2下1、右1下2、左1下2、左2下1）位置上的棋子与空格交换。求出最少交换次数使得棋盘变成目标局面。超过13步视为无解。

## H.2 Solution

深度优先搜索。每个状态记录棋盘当前局面和空格位置，枚举8个方向决策。初始ans设为14，如果当前局面为目标局面，则更新ans，并退回搜索的上一层，否则继续搜索。加一个剪枝技巧：如果当前局面不是目标局面，但当前步数$\geq$ ans+1，则不必继续搜索，退回搜索上一层（继续搜索更优）。

# I　等式

## I.1　Description

你有一个0-9的数字串，现在要从中插入一个加号和一个等号，使得插入后的串形如"a+b=c"，其中a,b,c都是无前导0的非空数字串，且数字a和数字b的和等于数字c。

保证答案一定存在。

## I.2　Solution

首先，设len(x) 表示数字x的位数，对于a + b = c 这个等式，假设len(a) ≥ len(b)，则len(a) 一定等于len(c) 或len(c) - 1。又因为len(a) + len(b) + len(c) = 数字串长N, 故在确定串c长度的情况下只有四种情况需要讨论。

关键是如何O(1) 判断等式是否成立。假设我们已知Ma = a % P，Mb = b % P, Mc = c % P，若a + b = c，则(Ma + Mb) % P = Mc；反过来若后者成立，则前者有很大概率成立（假设a，b 随机生成）。如果尝试多个模数P（一般使用质数），且后者都成立，则我们几乎可以肯定前者成立了。所以我们只用O(1) 计算一个串对应数字对某个质数的模：

对于数字串$s(1..N)$，质数$P$，设$h_i = (h_{i-1}*10+number(s_i)) \mod P$，则串$s[L..R] \mod P = (s_L * 10^{R-L} + ... + s_{R-1} * 10^1 + s_R) \mod P = (h_R - h_{L-1} * 10^{R-L+1}) \mod P$

注意判断前导0.

# J　斗地主

## J.1　Description

三个人斗地主，起始分均为0。

地主赢+6，两位农民−3；反之地主−6，两位农民+3。

单局得分可以翻整数次倍。

给定一组得分，问是否合法。

## J.2　Solution

首先忽略倍数的概念，任意倍的一局比赛都可以被若干次单局替换。

考虑三个人的最终分数为$x, y, z$，则首先我们有

$$\begin{cases} x \equiv y \equiv z \equiv 0 \mod 3 \\ x + y + z = 0 \end{cases} \tag{1}$$

其次考虑一共有三种得分情况，即三个人分别当地主，设三种情况的出现次数分别为$a, b, c$。

注意$a, b, c$ 均可以为负，代表地主失败的情况。则有

$$\begin{cases} 6a - 3b - 3c & = x \\ -3a + 6b - 3c & = y \\ -3a - 3b + 6c & = z \end{cases}$$

这个线性方程组的系数矩阵秩为2，进行简单的代换得到恒等式

$$a - b = \frac{x - y}{9}$$

由于$a - b$ 是整数，因此

$$9 \mid (x - y) \tag{2}$$

综上，满足式(1) 和(2) 的一组输入$x, y, z$ 合法。

# K 领口罩

## K.1 Description

一个离散区间上n个位置，初始有一些位置（k个）打上标记，接下来有q次两类操作：（1）在指定空位打标记;（2）找一个离最近打上标记的位置最远的空位打标记。

$n, k, q \leq 10^6$ 且$k + q \leq n$

## K.2 Solution

解法1：

用线段树维护区间内最小有标记位置、最大有标记位置、最长最左连续空位长度及位置。每次操作（1）相当于改点;操作（2）相当于先查询整个区间，再改点。注意位置1和位置n的特殊情况。

解法2：

用堆（priority_queue）维护所有连续空位长度及位置，同时用平衡树（set）维护所有已打标记位置。每次操作（1）在平衡树中查询左邻位和右邻位，向堆中插入两个新连续空位;每次操作（2）先查询最长最左连续空位（注意，需查询平衡树，若该空位中间有点，则该连续空位为过时数据，需要丢弃），再回到操作（1）。注意位置1和位置n的特殊情况。

# L Sakuzyo的头套

## L.1 Description

给定三维空间中不重合的8 个点，问是否构成立方体。

## L.2 Solution

对每个点判断，剩余7 个点中，距离最近的三个点，距离相等，且边所在的向量互相正交。

# M A ? B Problem

## M.1 Description

交互题，给你A和B以及结果校验器，要你判断操作符(+, -, *, Xor)是啥。

最多猜10次。

### M.2 Solution

思博题，交互题学习题。

## N Naive数模

### N.1 Description

给定P编码和D编码的定义，先区分输入的二进制串的编码类型，再进行正确的解码。本题考察了模拟、二进制、位运算等基础知识。

串长$\leq 10^5$

### N.2 Solution

模拟。将二进制串全部读进来，第一遍按P编码试解码，若出现绝对值大于等于32768的瞬时电平，说明是D编码。第二遍，按照第一遍确定的编码类型进行解码。注意到，P编码每16位瞬时电平正好是short类型，可以用位运算给short类型变量赋值，直接输出，而不必手工进行补码原码转换和进制转换。

## O 哆啦A梦的铜锣烧

### O.1 Description

给定$n$个数，要求分成$m$个组，且第$i$个组的最大值不大于第$i+1$个组的最小值，求每个组数之和最大值的最小值。

### O.2 Solution

这是一道二分题目，即答案存在二分性质。首先将$n$个数升序排序，然后二分答案，遍历$n$个数依次分组，如果和超过二分值，则新开一组，最后判断分出的组数与$m$的关系，判断该二分值是否可行。