

CCF 全国信息学奥林匹克联赛(NOIP2016)复赛模拟赛

提高组 day2

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名称	外星密码	挥金如土	量子谜题
英文题目与子目录名	password	dissipation	quantum
可执行文件名	password	dissipation	quantum
输入文件名	password.in	dissipation.in	quantum.in
输出文件名	password.out	dissipation.out	quantum.out
每个测试点时限	1s	1s	1s
测试点数目	10	10	10
每个测试点分值	10	10	10
结果比较方式	传统	传统	传统
题目类型	全文比较		
运行内存上限	128MB	128MB	128MB

二. 提交源程序文件名

对于 C++ 语言	password.cpp	dissipation.cpp	quantum.cpp
对于 C 语言	password.c	dissipation.c	quantum.c
对于 pascal 语言	password.pas	dissipation.pas	quantum.pas

三. 编译命令 (不包含任何优化开关)

对于 C++ 语言	g++ -o password password.cpp	g++ -o dissipation dissipation.cpp	g++ -o quantum quantum.cpp
对于 C 语言	gcc -o password password.c	gcc -o dissipation dissipation.c	gcc -o quantum quantum.c
对于 pascal 语言	fpc password.pas	fpc dissipation.pas	fpc quantum.pas

1.外星密码 (password.cpp/c/pas)

【问题描述】

小 W 潜入了外星人的基地刺探情报，遇到一个二级密码系统。一级密码是一个长度为 n 的 0-1 序列 B ，记为 $(b_1 \ b_2 \ \cdots \ b_n)$ 。将一级密码的第一位放到最后，得到一个新的序列 $(b_2 \ b_3 \ \cdots \ b_1)$ ，继续做同样的操作得到 $(b_3 \ b_4 \ \cdots \ b_2)$ ，如此反复，总共可以得到 n 个序列，将这些序列按字典序排序后，字典序最小的即为二级密码，输入这个二级密码，就可以得到情报。

外星人将领将排序过后的 n 个字符串按照字典序从小到大的顺序逐行写成了一个 n 阶矩阵，而且小 W 恰好看到了这个矩阵，不难发现，这个矩阵的第一行就是二级密码，所以如果小 W 记住这个矩阵的第一行，就可以直接获得情报。但是小 W 在出发前听上司布置任务时走神将“记住第一行”听成了“记住最后一列”。所以小 W 现在只记得最后一列是什么，小 W 赶紧联系远在地球的你，希望你能帮他由矩阵的最后一列得到矩阵的第一行。

【输入格式】

输入文件名为 password.in。

共两行，第一行一个整数 n ，第二行 n 个值为 0 或 1 的整数，表示矩阵的最后一列。

【输出格式】

输出文件名为 password.out。

输出文件共一行，为矩阵的第一行。

【样例】

password.in	password.out
5 1 1 0 0 0	0 0 1 0 1

【数据规模与约定】

对于 20% 的测试数据 $n \leq 20$

对于 50% 的测试数据 $n \leq 1000$

对于 100% 的测试数据 $n \leq 10000$

【样例解释】

一级密码为 $(0 \ 1 \ 0 \ 0 \ 1)$ ，因此得到 5 个序列为：
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$
，经过排序过后的

矩阵为：
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$
，第一行为 $(0 \ 0 \ 1 \ 0 \ 1)$ 。

2.挥金如土

(dissipation.cpp/c/pas)

【问题描述】

拉斯维加斯的赌场推出了一种新的游戏: 屏幕上有一个 n 位二进制数, 每一位将独立地随机产生, 产生 0 的概率为 p , 产生 1 的概率为 $1-p$, 你可以对 $0 \sim k-1$ 中的某个数字 x 下注, 如果最终产生的二进制数除以 k 的余数为 x , 那么你就赢了, 所有对 x 下注的玩家均以均分奖金, 而所有未对 x 下注的玩家将会输掉自己的赌注。

小 W 现在就在这个游戏机旁, 荷官问他如何下注。作为 (伪) 土豪, 自然地, 他想知道如果下注, 赢的概率最小是多少 (显然对不同的 x 下注赢钱的概率可能不一样), 不过小 W 心算能力实在太差, 于是他拜托你帮他评估一下赢钱的概率最小是多少。

【输入格式】

输入文件名为 `dissipation.in`。

输入文件共一行, 依次为空格隔开的 n, k, p , 其中 n, k 为正整数, p 为 $0 \sim 1$ 之间的浮点数。

【输出格式】

输出文件名为 `dissipation.out`。

输出文件共一行, 为小 W 赢钱的最小概率, 保留 3 位小数。

【样例】

dissipation.in	dissipation.out
3 8 0.5	0.125

【数据规模与约定】

对于 20% 的测试数据 $n \leq 20$

对于 50% 的测试数据 $n \leq 100$

对于 100% 的测试数据 $n \leq 2000000000, k \leq 100$

3.量子模型 (quantum.cpp/c/pas)

【问题描述】

小 W 所在学校的物理学水平全国闻名，因此吸引来了大量民间科学爱好者。某一天，一位自称能做出诺贝尔奖级别成果，颠覆人类物理学框架的民间科学爱好者拦住了小 W，和小 W 分享了一种小 W 闻所未闻的量子模型：

一个量子的状态可以用一个有序三元组来表示，如果该量子的状态是 (x,y,z) ，则能够发生如下六种之一的跃迁，转移到另一个状态（如果原状态 (x,y,z) 满足这种跃迁发生的前提条件的话）。

前提条件	跃迁后状态
$z < \min\{2y-x, x\}$ 或者 $z > \max\{2y-x, x\}$	$(2y-x, y, z)$
$z < \min\{2x-y, y\}$ 或者 $z > \max\{2x-y, y\}$	$(x, 2x-y, z)$
$y < \min\{2z-x, x\}$ 或者 $y > \max\{2z-x, x\}$	$(2z-x, y, z)$
$y < \min\{2x-z, z\}$ 或者 $y > \max\{2x-z, z\}$	$(x, y, 2x-z)$
$x < \min\{2y-z, z\}$ 或者 $x > \max\{2y-z, z\}$	$(x, y, 2y-z)$
$x < \min\{2z-y, y\}$ 或者 $x > \max\{2z-y, y\}$	$(x, 2z-y, z)$

所有的量子的初始状态都是 (i,j,k) ，且总满足 $i+k=2j$ ，现在对于任意的一个状态 (x,y,z) ，定义三元函数 $\text{dist}(x,y,z) = (i,j,k)$ 最少需要 $\text{dist}(x,y,z)$ 次跃迁可以变成状态 (x,y,z) 。这位民间科学爱好者同时声称，如果将一个量子一生中所有达到过的看作一个集合 S,那么 S 将满足：

1. $|S| = n$
 2. $\max\{\text{dist}(x,y,z) | (x,y,z) \in S\} = m$
- 他想知道有多少种不同的集合 S 满足上述条件。

【输入格式】

输入文件名为 quantum.in。

输入文件包含一行 5 个以空格隔开的整数：i j k n m

【输出格式】

输出文件名为 quantum.out。

输出文件包含一个整数，表示满足条件的集合 S 有多少种，为了方便起见，你只要输出这个数字对 2016 取模的结果。

【样例 1】

quantum.in	quantum.out
1 2 3 2 1	2

【样例 2】

quantum.in	quantum.out
1 2 3 3 2	4

【数据规模与约定】

对于 30%的测试数据 $n, m \leq 10$

对于 50%的测试数据 $m, n \leq 50$

对于 100%的测试数据 $m, n \leq 150, -50 \leq i, j, k \leq 50$