

正睿 23 NOIP 赛前 20 天 Day 03

| 题目名称 | 你 | 太 | 极端 | 了 |
|---------|---------|----------|-------------|---------|
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 英文题目名称 | you | very | extreme | end |
| 输入文件名 | you.in | very.in | extreme.in | end.in |
| 输出文件名 | you.out | very.out | extreme.out | end.out |
| 每个测试点时限 | 2s | 2s | 1s | 3s |
| 内存限制 | 512MB | 512MB | 512MB | 512MB |
| 提交的源文件名 | you.cpp | very.cpp | extreme.cpp | end.cpp |

【C++编译选项】 `-lm -std=c++14 -Wl,--stack=1000000000 -O2`

【试题下载地址】 <ftp://172.16.2.202/竞赛资料/20231106.zip> 匿名访问即可

【结果上传地址】 <ftp://172.16.2.202/20231106文件回收> 用户名密码均为test

【赛后补题地址】 <http://zhb.wms.edu/d/JH2023/>

【提交文件夹格式】

--准考证号\ (平时训练用中文姓名)

```
--task1\  
|      |--task1.cpp  
--task2\  
|      |--task2.cpp  
--task3\  
|      |--task3.cpp  
--task4\  
|      |--task4.cpp
```

T1 【题目名称】

你 (you)

【题目描述】

有个 $n \times m$ 的矩阵，行编号为 $0 \dots n-1$ ，列编号为 $0 \dots m-1$ ，第 i 行第 j 列一开始为 $im+j$.

现在支持三种操作：交换两行，交换两列，或者交换某两个位置。

求进行完 q 次操作后矩阵的形态。

【输入】

为了减小输入规模，本题采取以下方式读入：

第一行四个数 $n, m, q, seed$ 表示行数，列数，操作数，随机种子。

生成器代码如下：

```
uint64_t seed;
uint64_t next(){ //xorshift64
    seed ^= seed << 13;
    seed ^= seed >> 7;
    seed ^= seed << 17;
    return seed;
}
```

接下来一个长度为 q 的串 s ，表示 q 次操作分别的类型。

对于第 i 次操作，如果 s_i 是 r ，则操作为交换第 r_1, r_2 行，如果是 c 则为交换第 c_1, c_2 列， f 则为交换 $(r_1, c_1), (r_2, c_2)$ ，这些变量按顺序分别是调用`next()`的返回值对 n 或 m 取模后的值。

【输出】

为了减小输出规模，设 $a_{i,j}$ 为最后矩阵第 i 行第 j 列的值，只需要输出 $((\sum_{i,j} a_{i,j} 17^i 19^j) \bmod 998244353)$ 之后的值。

【样例】

3 5 3 114514
crf

382914571

【样例解释】

操作序列为：

c 3 0
r 2 1
f 1 2 2 2

最终矩阵为：

3 1 2 0 4
13 11 7 10 14
8 6 12 5 9

【样例二】

见下发文件。

【数据范围】

对于100%的数据， $1 \leq n, m \leq 5000, 1 \leq q \leq 10^6, 0 \leq seed < 2^{64}$.

| 子任务编号 | $n, m \leq$ | $q \leq$ | 特殊性质 | 分值 |
|-------|-------------|----------|-----------|----|
| 1 | 100 | 100 | | 20 |
| 2 | 5000 | 5000 | | 20 |
| 3 | 5000 | 10^6 | s 中没有 f | 30 |
| 4 | 5000 | 10^6 | | 30 |

T2 【题目名称】

有点太 (very)

【题目描述】

有 n 个砝码，根据材质不同质量只有1g,2g,3g三种。

现在砝码上的质量标签都遗失了，由于只有材质不同，从外表难以分辨。但所幸还有一个天平，可以用这个天平称量砝码之间的重量关系。

某些砝码之间的重量关系已经称出来了，但其它的还不知道。

现在已经选了两个放在了天平一边，想再选两个放在另一边，求有多少种选法使得由已知信息能确定左边更重/相等/右边更重。

【输入】

第一行一个数 n 。

接下来 n 行每行 n 个字符，包含 + / - = / ? ,第 i 行第 j 个字符 $s_{i,j}$ 为 + 表示第 i 个比第 j 个大，- 表示更小，= 表示相等，? 表示不知道。

保证 $s_{i,i}$ 是 ? , 且如果 $s_{i,j}$ 是 + / - = / ? 有 $s_{j,i}$ 是 - / + / = / ? 。

接下来两个数 a,b 表示放在左边的编号。

保证存在至少一种质量序列满足所有称量结果。

【输出】

三行，每行一个数，按顺序分别有多少种选法使得由已知信息能确定左边更重/相等/右边更重。

【样例】

6
?+????
-?+???
?-????
????+?
???-?+
????-?
2
5

1
4
1

【数据范围】

对于100%的数据， $n \leq 50$ 。

| 子任务编号 | $n \leq$ | 特殊性质 | 分值 |
|-------|----------|------|----|
| 1 | 10 | | 30 |
| 2 | 20 | | 30 |
| 3 | 50 | 没有 = | 20 |
| 4 | 50 | | 20 |

T3 【题目名称】

极端 (extreme)

【题目描述】

城市里有一条直路，两边各有n家店。从其中一个端点往另一个端点数，其中一边的第i家店每天有 a_i 个顾客，另一边的第i家店每天有 b_i 个顾客，且这两家店正好面对面。(认为这些顾客两两不同)

现在这 $2n$ 家店共同的母公司想在这条路上投放一个新的广告。它可以将广告贴在某些店的门上，这样另一侧的两侧最多 d 家店里的顾客都能透过玻璃门看到这个广告，但是由于广告是不透明的，这家店的顾客就看不到外面的广告了。

求一个贴广告的方案，使得能看到至少一条广告为顾客数最多。

形式化的，设这条路从 $(1,0)$ 到 $(n,0)$ ，则店铺排列在 $(i,1)(i,-1)$ 位置 $(1 \leq i \leq n)$ 。 $(i,1)$ 位置的店有 a_i 名顾客， $(i,-1)$ 位置的店有 b_i 名顾客。

可以选择一些店铺贴上广告，一家店 (x,y) 里的顾客能看到广告的条件为 (x,y) 没贴广告，且存在 $|x'-x| \leq d$ 使得 $(x',-y)$ 门口贴了广告，最大化能从店里看到广告为顾客数。

【输入】

第一行两个数 n,d 。

第二行 n 个数 a_1, \dots, a_n 。

第三行 n 个数 b_1, \dots, b_n 。

【输出】

一个数，表示能从店里看到广告为顾客数的最大值

【样例】

5 2
1 2 2 4 3
5 1 3 2 2

20

【样例二】

见下发文件。

【数据范围】

对于100%的数据, $0 \leq d < n \leq 1500, 1 \leq a_i, b_i \leq 10^5$.

| 子任务编号 | $n \leq$ | 特殊性质 | 分值 |
|-------|----------|------------|----|
| 1 | 10 | | 20 |
| 2 | 50 | | 20 |
| 3 | 300 | | 20 |
| 4 | 1500 | $d \leq 5$ | 20 |
| 5 | 1500 | | 20 |

T4 【题目名称】

了 (end)

【题目描述】

在《计算机系统概论》这门课上接触了指令集之后，小W造了一个非常简易的计算机系统。

这个系统有四个8位(即值域0..255)的寄存器 ax, bx, cx, dx ，指令只包含or/and/xor位运算，分为两类：

- 1. 两个都是寄存器(or $r1\ r2$ /and $r1\ r2$ /xor $r1\ r2$)，会将 $r1$ 寄存器中的值变为两个寄存器的值 or/and/xor 之后的结果，如 or $ax\ bx$ ($r1$ 和 $r2$ 可以是同一个寄存器)
 - 2. 其中一个是数值(ori $r1\ x$ /andi $r1\ x$ /xori $r1\ x$)，会将 $r1$ 寄存器中的值 or/and/xor 上 x ，例如 ori $ax\ 114$
- 这两种指令每个都需要一个系统时钟周期。

与此同时，这个系统支持一个简单的循环，只需要用 repeat k 和 end 包裹中间的命令，系统就会对将中间的指令按顺序重复 k 次，其中 $2 \leq k \leq 255$ ，且不需要额外时间(即每次执行位运算次数的时候才需要时间).

为了检验这个系统有没有错误，小W写下了一段共 n 行的指令代码，并且进行了 q 次测试，每次给定 ax, bx, cx, dx 的初值，并让系统从头开始执行了 k 个时钟周期的时间，然后看看此时四个寄存器的值分别是多少。由于他没空计算正确的值了，所以想请你帮他算一下四个寄存器应该是多少，以方便他的测试。

保证代码符合语法，且运行完代码至少需要k个时钟周期的时间

【输入】

第一行两个数n,q，表示行数和测试次数。

接下来n行，每行形如 or/and/xor r1 r2 或者 ori/andi/xori r1 x 或者 repeat k 或者 end，表示代码的一行，具体参考样例

接下来q行每行四个数 k,ax,bx,cx,dx，表示一次测试的初值和运行时长。

【输出】

q行每行四个数 ax,bx,cx,dx 表示四个寄存器的值。

【样例】

```
6 2
repeat 5
xor ax bx
xori ax 3
and cx ax
xor cx dx
end
10 1 2 4 3
8 4 1 2 3
```

```
0 2 2 3
4 1 3 3
```

【样例二】

见下发文件。

【数据范围】

对于100%的数据, $n \leq 12000, q \leq 10000$.

| 子任务编号 | $n \leq$ | $q \leq$ | 特殊性质 | 分值 |
|-------|----------|----------|----------|----|
| 1 | 5 | 10 | | 10 |
| 2 | 20 | 20 | | 10 |
| 3 | 2000 | 2000 | | 25 |
| 4 | 12000 | 10000 | 保证循环没有嵌套 | 15 |
| 5 | 12000 | 10000 | 保证循环一定嵌套 | 15 |
| 6 | 12000 | 12000 | | 25 |

勘误：询问的 $k \leq 10^9$