NOIP 模拟赛

PYWBKTDA

中国象棋（chess）

题目描述

小 A 在玩象棋，象棋分为红黑双方，其规则如下：

棋盘： 坐标集合 $\{(x,y) \mid x \in [1,10] \cap Z, y \in [1,9] \cap Z\}$

红方河界： 坐标集合 $\{(x,y) \mid x \in [1,5] \cap Z, y \in [1,9] \cap Z\}$

黑方河界： 坐标集合 $\{(x,y) \mid x \in [6,10] \cap Z, y \in [1,9] \cap Z\}$

红方九宫： 坐标集合 $\{(x,y) \mid x \in [1,3] \cap Z, y \in [4,6] \cap Z\}$

黑方九宫： 坐标集合 $\{(x,y) \mid x \in [8,10] \cap Z, y \in [4,6] \cap Z\}$

棋子： 每个棋子属于红黑双方中的一方且位于棋盘内两两不同的坐标上

棋子可以移动，移动后的位置上需无己方棋子，且若为对方棋子则将其替换为己方所移动的棋子

棋子共分为 7 类，每类棋子的移动规则如下：

将 (j)： 每方恰有一个，位于己方九宫内，移动时从 (x,y) 到 (u,v) 满足以下条件之一：

- $|x-u|+|y-v|=1$ 且 (u,v) 也位于己方九宫内
- $y=v,(u,v)$ 上为对方的将且 $\forall(z-x)(z-u)<0,(z,y)$ 上无棋子

士 (s)： 每方至多两个，位于己方九宫内，移动时从 (x,y) 到 (u,v) 满足 $|x-u|=|y-v|=1$ 且 (u,v) 也位于己方九宫内

相 (x)： 每方至多两个，位于己方河界内，移动时从 (x,y) 到 (u,v) 满足 $|x-u|=|y-v|=2$、$(\frac{x+u}{2},\frac{y+v}{2})$ 上无棋子且 (u,v) 也位于己方河界内

车 (c)： 每方至多两个，移动时从 (x,y) 到 (u,v) 满足以下条件之一：

- $x=u,y\neq v$ 且 $\forall(z-y)(z-v)<0,(x,z)$ 上无棋子
- $y=v,x\neq u$ 且 $\forall(z-x)(z-u)<0,(z,y)$ 上无棋子

马 (m)： 每方至多两个，移动时从 (x,y) 到 (u,v) 满足以下条件之一：

- $|x-u|=1,|y-v|=2$ 且 $(x,\frac{y+v}{2})$ 上无棋子
- $|x-u|=2,|y-v|=1$ 且 $(\frac{x+u}{2},y)$ 上无棋子

炮 (p)： 每方至多两个，移动时从 (x,y) 到 (u,v) 满足以下条件之一：

- (u,v) 上无棋子且满足以下条件之一：
  - $x=u, y\neq v$ 且 $\forall (z-y)(z-v)<0, (x,z)$ 上无棋子
  - $y=v, x\neq u$ 且 $\forall (z-x)(z-u)<0, (z,y)$ 上无棋子
- (u,v) 上为对方棋子且满足以下条件之一：
  - $x=u, y\neq v$ 且 $\forall (z-y)(z-v)<0, (x,z)$ 上恰有 1 个棋子
  - $y=v, x\neq u$ 且 $\forall (z-x)(z-u)<0, (z,y)$ 上恰有 1 个棋子

兵 (b)： 每方至多 5 个，移动时从 (x,y) 到 (u,v) 满足以下条件之一：

- $x=u-1, y=v$ 且该兵属于红方
- $x=u+1, y=v$ 且该兵属于黑方
- $x=u, |y-v|=1$ 且 (x,y) 位于对方河界内

小 A 想写一个象棋 AI，他很快想到了搜索算法，为了评估这个算法的运行效率，他需要求出局面中双方所有棋子的移动方式之和

输入格式

从文件 chess.in 中读入数据

第一行一个整数 t，表示数据组数

接下来 t 组数据，每组数据为 10 行 9 列的字符矩阵，第 i 行第 j 列的字符表示 (i,j) 上的棋子（分别用棋子后括号内字母的小写和大写形式表示红方和黑方的该棋子，若不存在棋子则为 *）

输出格式

输出到文件 chess.out 中

共 t 行，每行一个整数，表示双方所有棋子的移动方式之和

样例输入

1
cmxsjsxmc
****
*p*****p*
b*b*b*b*b
****
****
B*B*B*B*B
*P*****P*
****
CMXSJSXMC

样例输出

88

样例解释

- 每个将/士/兵有一种移动方式
- 每个相/车/马有两种移动方式
- 每个炮有 12 种移动方式

因此，答案为 $1 \times (2+4+10) + 2 \times (4+4+4) + 12 \times 4 = 88$
更多样例见下发文件 chess.in 和 chess.ans

数据范围

对于所有测试点，保证 $1 \leq t \leq 100$，局面满足规则中的条件

测试点编号	输入的矩阵字符集
1	$\subseteq \{*, j, J\}$
2	$\subseteq \{*, j, J, s, S, x, X\}$
3	$\subseteq \{*, j, J, s, S, x, X, c, C\}$
4	$\subseteq \{*, j, J, s, S, x, X, c, C, m, M\}$
5,6	$\subseteq \{*, j, J, s, S, x, X, c, C, m, M, p, P\}$
7,8	$\subseteq \{*, j, J, s, S, x, X, c, C, m, M, p, P, b, B\}$
9,10	$\subseteq \{*, j, J, s, S, x, X, c, C, m, M, p, P, b, B\}$

时间限制： 1s

空间限制： 512MB

树（tree）

题目描述

小 A 在玩一个游戏，游戏中他位于一棵 n 个点的树上

小 A 每次可以选择结束游戏或将移动到相邻的位置上，如果在节点 i 上结束游戏，则可以获得 a_i 的分数

显而易见，小 A 应该移动到分数最大的位置，但这个游戏有一个 bug，每次移动后会有一个除了他当前所在位置以外的节点消失

小 A 是一个谨慎的人，即需要对每一个 i，求出当小 A 初始位于 i 时，最坏情况下他使用最优策略所能获得的分数

提示：小 A 的最优策略可以取决于之前消失的节点，最坏情况下消失的节点也可以取决于之前小 A 的移动

输入格式

从文件 tree.in 中读入数据

第一行一个整数 n，表示树的大小

第二行 n 个整数 $a_1, a_2, ..., a_n$，表示节点 i 的分数

接下来 $n-1$ 行，每行两个整数 x, y，表示树上的一条边 (x, y)

输出格式

输出到文件 tree.out 中

一行 n 个整数，第 i 个整数表示初始位于 i 时的答案

样例输入

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

样例输出

4 5 2 5 4

样例解释

当小 A 以 1 为起点时，最优策略是移动到 2，最坏情况下 4 消失，此时小 A 结束游戏，所得分数为 4

当小 A 以 2 为起点时，最优策略是移动到 4，不论哪个节点消失，此时小 A 结束游戏，所得分数为 5

当小 A 以 3 为起点时，最优策略是直接结束游戏（若移动到 1，则最坏情况下节点 2 消失），所得分数为 2

当小 A 以 4 为起点时，最优策略是直接结束游戏，所得分数为 5

当小 A 以 5 为起点时，最优策略是移动到 2，最坏情况下 4 消失，此时小 A 结束游戏，所得分数为 4

更多样例见下发文件 tree1/2/3.in 和 tree1/2/3.ans

数据范围

对于所有测试点，保证 $1 \le n \le 5 \times 10^5$，$1 \le a_i \le n$

测试点编号	n	特殊性质
1	≤ 18	无
2,3,4	$\le 10^3$	无
5	$\le 5 \times 10^5$	第 i 条边为 $(i, i+1)$
6,7	$\le 5 \times 10^5$	$a_i \in \{1, 2\}$
8,9,10	$\le 5 \times 10^5$	无

时间限制： 1s

空间限制： 512MB

操作序列（operate）

题目描述

小 A 有一个长为 n 的序列 a，他设计了 m 个操作，每个操作形如：

- 1 x y 表示交换 a_x 和 a_y
- 2 l r z 表示修改 $\forall i \in [l,r], a_i = z$
- 3 z 表示查询 a_z

这些操作实在太简单了，于是小 A 又给出 q 个区间 $[L,R]$，求初始序列 a 全为 0 且依次执行 $[L,R]$ 内的操作后，所有 3 操作的答案之和

输入格式

从文件 operate.in 中读入数据
第一行三个整数 n,m,q，表示序列长度、操作数和询问数
接下来 m 行，每行一个操作，具体格式见题目描述
接下来 q 行，每行两个整数 L,R，表示一组询问

输出格式

输出到文件 operate.out 中
共 q 行，每行一个整数，表示答案

样例输入

```
3 6 3
2 1 3 1
3 3
2 2 3 2
1 1 2
3 1
3 2
1 6
3 6
2 5
```

样例输出

4
2
2

样例解释

第一组询问答案为 $1+2+1=4$
第二组询问答案为 $2+0=2$
第三组询问答案为 $0+2=2$
更多样例见下发文件 operate1/2.in 和 operate1/2.ans

数据范围

对于所有测试点，保证 $1 \le n, m, q \le 10^6$
$1 \le x, y, z \le n$，$x \ne y$，$1 \le l \le r \le n$，$1 \le L \le R \le m$

测试点编号	n	m, q	特殊性质
1,2	$\le 10^4$	$\le 10^3$	无
3,4,5	$\le 5 \times 10^4$	$\le 5 \times 10^4$	无
6,7	$\le 10^6$	$\le 10^6$	不存在操作 1
8,9,10	$\le 10^6$	$\le 10^6$	无

时间限制： 1s
空间限制： 512MB
提示： 建议使用较快的 IO 方式，可参考下发文件 IO.cpp

集合划分（partition）

题目描述

小 A 在思考一个问题：将 n 个数划分为 m 个集合，求所有划分方案中，每个集合元素和的乘积和

但小 A 不会这个问题，于是他想到了与加法类似的运算——异或，并将其中的对集合元素的求和改成了异或

同时，小 A 很喜欢空集，但这样的划分方案中如果出现空集对应的结果一定为 0，因此他还在之前的基础上把每个集合的结果 +1

具体而言，定义 $A \sqsubseteq S$ 表示 A 为 S 的划分，即

$$\begin{cases} \bigcup_{T \in A} T = S \\ \forall X, Y \in A, X \neq Y, X \cap Y = \emptyset \end{cases}$$

给定长为 n 的序列 a，你需要求出

$$\sum_{\{S_1, S_2, \ldots, S_m\} \sqsubseteq [1,n]} \prod_{i=1}^{m} \left(1 + \bigoplus_{j \in S_i} a_j\right)$$

由于答案可能非常大，只需要求出对 $10^9 + 7$ 取模后的结果

输入格式

从文件 partition.in 中读入数据

第一行两个整数 n 和 m，表示序列长度和划分的集合数

第二行 n 个整数 $a_1, a_2, ..., a_n$，表示序列 a

输出格式

输出到文件 partition.out 中

一行一个整数，表示答案

样例输入

3 2

1 2 3

样例输出

60

样例解释

答案为

$$2\cdot(1\cdot(1+1\oplus 2\oplus 3)+(1+1)\cdot(1+2\oplus 3)+(1+2)\cdot(1+1\oplus 3)+(1+3)\cdot(1+2\oplus 3))=60$$

更多样例见下发文件 partition1/2/3.in 和 partition1/2/3.ans

数据范围

对于所有测试点，保证 $1\le n\le 16$，$1\le m\le 10^9$，$0\le a_i<2^{30}$

测试点编号	n	m	特殊性质
1	≤ 10	≤ 10	无
2	≤ 13	≤ 13	无
3,4	≤ 13	$\le 10^9$	无
5,6	≤ 16	$\le 10^9$	$a_i=2^i$
7,8,9,10	≤ 16	$\le 10^9$	无

时间限制： 500ms

空间限制： 512MB