

# NOI2024 联合省选 Day2

时间：2024 年 2 月 23 日

题目名称	puzzle	witch	couple
题目类型	传统题	传统题	传统题
目录	puzzle	witch	couple
可执行文件名	puzzle	witch	couple
输入文件名	puzzle.in	witch.in	couple.in
输出文件名	puzzle.out	witch.out	couple.out
每个测试点时限	2 秒	3 秒	2 秒
内存限制	256 MB	512 MB	1024 MB
子任务数目	6	5	20
测试点是否等分	否	否	是

提交源程序文件名

对于 C++ 语言	puzzle.cpp	witch.cpp	couple.cpp
-----------	------------	-----------	------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 只提供 Linux 格式附加样例文件。
9. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

puzzle (puzzle)

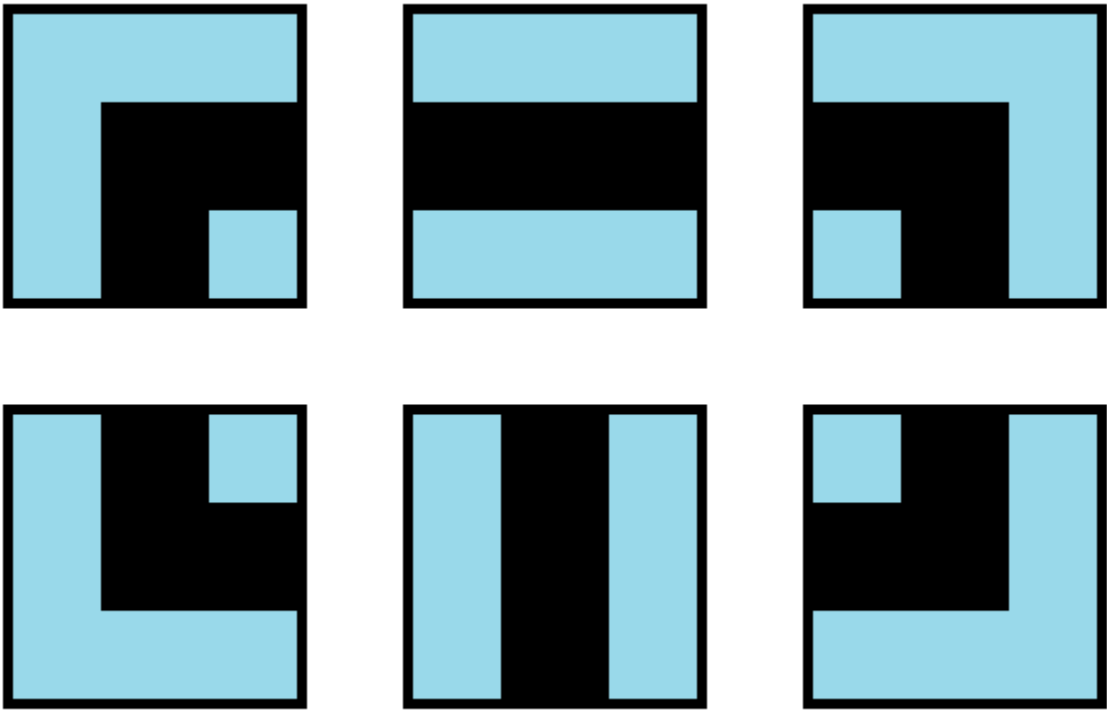
【题目描述】

YYX 转职成了 puzzle master!

loopy 是一个风靡全球的 puzzle, YYX 对 loopy 研究极深, 并受其启发开发了一款新的游戏:

给你一个  $n \times m$  的网格, 其中有若干个格子标记为 0, 意味着你必须往上面放一个管道; 剩余的格子标记为 1, 意味着上面不能放管道。

管道有如下六种:



与 loopy 相同的是, 你需要使得你放置的管道要形成闭合回路, 不同的是, 形成的闭合回路可以不止一个。

可以发现, 在这种限制下, 只去找到是否有解对于 puzzle master 来说实在太简单了, 于是 YYX 又给网格上每个格子都赋了一个权值。一种放置管道方案的成绩定义为, 在这个方案中, 放置有 L 型管道 (即上图中第 1,3,4,6 种管道) 的那些格子的权值之和。YYX 要求, 假如至少有一种放置管道的方案, 则需要找到其中成绩最大的方案。

作为 puzzle master, YYX 当然想解决这个新问题。但是她太笨了, 所以她找到了你,

希望你帮她解决问题。

【输入格式】

从文件 `puzzle.in` 中读入数据。  
第一行两个整数  $n, m$ 。  
接下来  $n$  行，每行  $m$  个整数分别表示第  $i$  行第  $j$  列的格子上的标记  $a_{i,j}$ 。  
接下来  $n$  行，每行  $m$  个整数分别表示第  $i$  行第  $j$  列的格子的权值  $w_{i,j}$ 。

【输出格式】

输出到文件 `puzzle.out` 中。  
假如至少存在一个放管道的方案，输出一个整数表示最大成绩；假如无解，输出 `-1`。

【样例 1 输入】

```
1 3 3
2 1 1 1
3 1 0 0
4 1 0 0
5 48 94 1
6 78 78 81
7 1 12 60
```

【样例 1 输出】

```
1 231
```

【样例 1 解释】

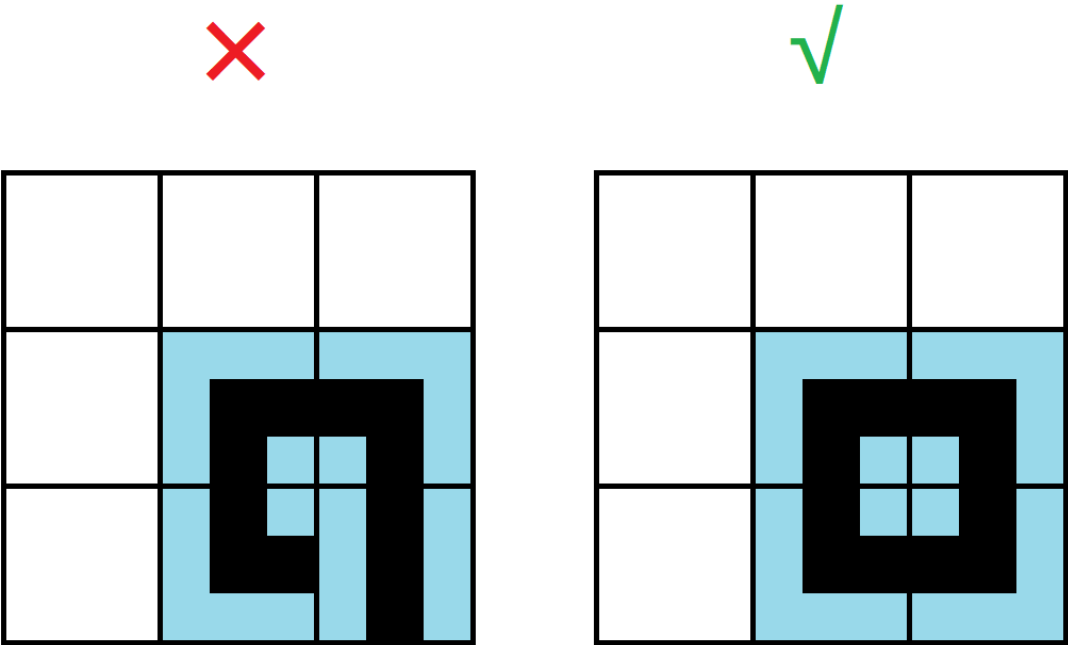
若下图，左图方案不合法，因为其没有构成闭合回路；右图是唯一一种合法方案，成绩为  $78 + 81 + 12 + 60 = 231$ 。

【样例 2】

见选手目录下的 `puzzle/puzzle2.in` 与 `puzzle/puzzle2.ans`。

【样例 3】

见选手目录下的 `puzzle/puzzle3.in` 与 `puzzle/puzzle3.ans`。



【样例 4】

见选手目录下的 `puzzle/puzzle4.in` 与 `puzzle/puzzle4.ans`。

【数据范围】

- 对于全部的数据， $1 \leq n \leq 150, 1 \leq m \leq 30, 0 \leq w_{i,j} \leq 100, a_{i,j} \in \{0, 1\}$ 。
- subtask1 11pts,  $n = m = 3$ 。
  - subtask2 19pts,  $n = 20, m = 6$ 。
  - subtask3 24pts,  $n = 50, m = 14$ 。
  - subtask4 16pts,  $m \leq 20, w_{i,j} = 0$ 。
  - subtask5 15pts,  $m \leq 20, w_{i,j} \leq 1$ 。
  - subtask6 14pts, 无特殊限制。

## witch (witch)

### 【题目描述】

YYX 转职成了巫医！

邻国的军队正好发生了瘟疫，于是国王邀请她作为国师来为军队治疗。

军队有  $n$  个士兵排成一排，从左到右第  $i$  个士兵的武力值为  $a_i$ ，初始时没有士兵得瘟疫。

接下来  $m$  天，每天早上会发生如下四种事件之一：

- YYX 为军队祈福：从左到右第  $l_i$  个到第  $r_i$  个士兵的武力值增加  $k_i$ 。
- 国王进行点兵：向 YYX 询问从左到右第  $l_i$  个到第  $r_i$  个士兵中，没有得瘟疫的士兵的武力值之和。
- 敌国的巫师散播瘟疫：第  $pos_i$  个士兵得了瘟疫。
- YYX 为军队治疗：把从左到右第  $l_i$  个到第  $r_i$  个士兵中患有瘟疫的士兵治好。大病初愈的士兵非常虚弱，武力值会变为 0。注意，原本没有得瘟疫的士兵被治疗不会有任何影响。

每天晚上，瘟疫会开始蔓延，具体的，假如从左到右第  $i$  个士兵在这天晚上前就患有瘟疫，那么第  $i - 1$  个和第  $i + 1$  个士兵也会得瘟疫。

YYX 对于祈福和治疗得心应手，对数数一窍不通，所以她找到了你，希望你帮她解决问题。

### 【输入格式】

从文件 **witch.in** 中读入数据。

第一行两个整数  $n, m$ 。

第二行  $n$  个整数表示  $a_i$ 。

接下来  $m$  行，每行第一个正整数  $op_i$  表示第  $i$  天早上发生的事件种类。

当  $op_i = 1$ ，表示 YYX 进行祈福，后面紧跟三个整数  $l_i, r_i, k_i$ 。

当  $op_i = 2$ ，表示国王进行点兵，后面紧跟两个整数  $l_i, r_i$ 。

当  $op_i = 3$ ，表示敌国巫师散播瘟疫，后面紧跟一个整数  $pos_i$ 。

当  $op_i = 4$ ，表示 YYX 进行治疗，后面紧跟两个整数  $l_i, r_i$ 。

### 【输出格式】

输出到文件 **witch.out** 中。

对于国王每次点兵，回答一行一个整数表示答案。

【样例 1 输入】

```
1 6 6
2 1 2 3 4 5 6
3 3 4
4 2 1 3
5 4 2 5
6 1 2 6 3
7 2 1 5
8 2 1 6
```

【样例 1 输出】

```
1 3
2 7
3 4
```

【样例 1 解释】

记  $b_i$  为第  $i$  个人是否得了瘟疫。

第 1 天早上,  $a_i = [1, 2, 3, 4, 5, 6], b_i = [0, 0, 0, 1, 0, 0]$ 。

第 1 天晚上,  $a_i = [1, 2, 3, 4, 5, 6], b_i = [0, 0, 1, 1, 1, 0]$ 。

第 2 天早上,  $a_i = [1, 2, 3, 4, 5, 6], b_i = [0, 0, 1, 1, 1, 0]$ , 回答  $1 + 2 = 3$ 。

第 2 天晚上,  $a_i = [1, 2, 3, 4, 5, 6], b_i = [0, 1, 1, 1, 1, 1]$ 。

第 3 天早上,  $a_i = [1, 0, 0, 0, 0, 6], b_i = [0, 0, 0, 0, 0, 1]$ 。

第 3 天晚上,  $a_i = [1, 0, 0, 0, 0, 6], b_i = [0, 0, 0, 0, 1, 1]$ 。

第 4 天早上,  $a_i = [1, 3, 3, 3, 3, 9], b_i = [0, 0, 0, 0, 1, 1]$ 。

第 4 天晚上,  $a_i = [1, 3, 3, 3, 3, 9], b_i = [0, 0, 0, 1, 1, 1]$ 。

第 5 天早上,  $a_i = [1, 3, 3, 3, 3, 9], b_i = [0, 0, 0, 1, 1, 1]$ , 回答  $1 + 3 + 3 = 7$ 。

第 5 天晚上,  $a_i = [1, 3, 3, 3, 3, 9], b_i = [0, 0, 1, 1, 1, 1]$ 。

第 6 天早上,  $a_i = [1, 3, 3, 3, 3, 9], b_i = [0, 0, 1, 1, 1, 1]$ 。回答  $1 + 3 = 4$ 。

第 6 天晚上,  $a_i = [1, 3, 3, 3, 3, 9], b_i = [0, 1, 1, 1, 1, 1]$ 。

【样例 2】

见选手目录下的 `witch/witch2.in` 与 `witch/witch2.ans`。

**【样例 3】**

见选手目录下的 *witch/witch3.in* 与 *witch/witch3.ans*。

**【数据范围】**

对于全部的数据,  $1 \leq n, a_i, k_i \leq 10^5, 1 \leq m \leq 5 \times 10^5, 1 \leq pos_i \leq n, 1 \leq l_i \leq r_i \leq n$ 。

subtask1 19pts,  $n, m \leq 1000$ 。

subtask2 16pts,  $m \leq 10^5, op_i \leq 3$ 。

subtask3 15pts,  $n, m \leq 50000$ 。

subtask4 20pts,  $l_i, r_i, pos_i$  都在  $[1, n]$  中等概率随机生成。特别的, 如果此时存在  $l_i > r_i$ , 交换它们。

subtask5 30pts, 无特殊限制。

## couple (couple)

### 【题目描述】

YYX 转职成了爱神！

在 YYX 管辖的城市，有  $n$  个房子和  $n - 1$  个道路，每个道路连接着两个房子，保证任意两个房子可以只通过道路互相到达。其中，编号为 1 到  $m$  的房子各住着一个单身汉，其中，第  $i$  个单身汉的奇异值为  $a_i$ ，保证这些房子只与一条道路相连。对于编号为  $m + 1$  到  $n$  的房子，这些房子是公共建筑，保证这些房子分别恰好与三条道路相连。保证  $m$  是偶数。

善良的 YYX 希望给这些单身汉配对为情侣，考虑到情侣约会时如果被别人撞见会很尴尬，所以，她希望对于任意两对情侣  $(x, y), (x', y')$ ， $x$  到  $y$  的最短路径与  $x'$  到  $y'$  的最短路径不包含同一条道路。YYX 认为情侣应该互补，故一对情侣的匹配度定义为这两个人的奇异值的按位异或，一组配对方案的幸福度为这组匹配里所有情侣的匹配度之和。

容易发现，如果希望恰好匹配  $m/2$  对情侣，那么配对的方案是唯一存在的。但不巧的是，YYX 听说 Dr. Wu 也住在这个城市，她当然不希望他找到情侣，但 YYX 并不知道 Dr. Wu 具体住在哪一个房子。于是，她希望对于每一个  $i = 1, 2, \dots, m$ ，求出不包含第  $i$  个单身汉时，所有匹配了  $m/2 - 1$  对情侣的匹配方案中，最大的幸福度是多少。

但是她太笨了，所以她找到了你，希望你帮她解决问题。

### 【输入格式】

从文件 `couple.in` 中读入数据。

第一行两个整数  $n, m$ 。

第二行  $m$  个整数表示  $a_i$ 。

接下来  $n - 1$  行，每行两个整数  $u_i, v_i$ ，表示第  $i$  条道路连接了第  $u_i$  个房子和第  $v_i$  个房子。

### 【输出格式】

输出到文件 `couple.out` 中。

输出  $m$  个整数表示答案。



【样例 1 输入】

```
1 10 6
2 0 1 2 2 3 1
3 1 7
4 5 7
5 2 9
6 7 9
7 9 10
8 6 10
9 10 8
10 3 8
11 8 4
```

【样例 1 输出】

```
1 5
2 6
3 6
4 6
5 4
6 6
```

【样例 1 解释】

删掉第 1 个人后的最优配对是：(2, 5), (4, 6)，答案为  $a_2 \oplus a_5 + a_4 \oplus a_6 = 5$ 。  
删掉第 2 个人后的最优配对是：(1, 5), (4, 6)，答案为  $a_1 \oplus a_5 + a_4 \oplus a_6 = 6$ 。  
删掉第 3 个人后的最优配对是：(1, 5), (4, 6)，答案为  $a_1 \oplus a_5 + a_4 \oplus a_6 = 6$ 。  
删掉第 4 个人后的最优配对是：(1, 5), (3, 6)，答案为  $a_1 \oplus a_5 + a_3 \oplus a_6 = 6$ 。  
删掉第 5 个人后的最优配对是：(1, 2), (3, 6)，答案为  $a_1 \oplus a_2 + a_3 \oplus a_6 = 4$ 。  
删掉第 6 个人后的最优配对是：(1, 5), (2, 3)，答案为  $a_1 \oplus a_5 + a_2 \oplus a_3 = 6$ 。

【样例 2】

见选手目录下的 *couple/couple2.in* 与 *couple/couple2.ans*。

**【样例 3】**

见选手目录下的 *couple/couple3.in* 与 *couple/couple3.ans*。

**【样例 4】**

见选手目录下的 *couple/couple4.in* 与 *couple/couple4.ans*。

**【数据范围】**

对于 15% 的数据,  $n \leq 300$ 。

对于 35% 的数据,  $n \leq 5000$ 。

对于另 25% 的数据, 保证存在整数  $l$  使得  $n = 2^l - 2, m = n/2 + 1$ , 且城市形如将一个完全二叉树, 把根删去后两个儿子相连所成的结构。

对于 100% 的数据,  $6 \leq n \leq 3 \times 10^5, 1 \leq m \leq n \wedge m \equiv 0 \pmod{2}, 0 \leq a_i < 2^{12}$ 。