

元旦欢乐赛

时间：2023 年 12 月 30 日 8 : 00 ~ 12 : 00

题目名称	树上异或	见字如面	加了个加	过桥米线
题目类型	传统题	传统题	传统题	传统题
目录	xor	pair	number	noodle
可执行文件名	xor	pair	number	noodle
输入文件名	xor.in	pair.in	number.in	noodle.in
输出文件名	xor.out	pair.out	number.out	noodle.out
每个测试点时限	2.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	6	25	6
测试点是否等分	是	否（捆绑测试）	是	否（捆绑测试）

提交源程序文件名

对于 C++ 语言	xor.cpp	pair.cpp	number.cpp	noodle.cpp
-----------	---------	----------	------------	------------

编译选项

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

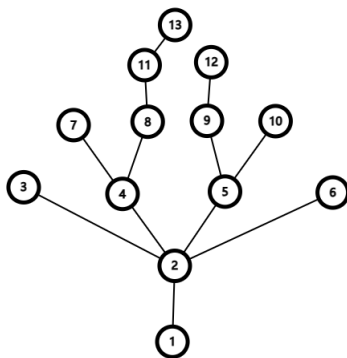
【注意事项（请仔细阅读）】

1. 选手提交的源程序必须存放在子文件夹中，文件夹名称与对应试题英文名一致。
2. 文件名（程序名和输入输出文件名）必须使用英文小写。
3. 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。
4. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
5. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 只提供 Windows 格式附加样例文件，同时数据在 Windows 下生成。
8. 每道题目所提交的代码文件大小限制为 100KB。
9. 评测在 Windows 下进行，统一评测时使用的评测机配置为 Intel(R) Core(TM) i5-8500 @ 3.00GHz，内存 8 GiB。

树上异或 (xor)

【题目背景】

所以 OIer 的圣诞树是这样的吗？



【题目描述】

小 N 做了一棵巨大的圣诞树。树上有 n 个结点，其中 1 号节点是根结点，除 1 号点外的所有结点都有父结点 f_i 。树上每个结点上有数字，数字均为 0 或 1。

每次操作你可以选择一个结点，将其子树内（包括自己）的结点上的所有数字异或上 1，即将 0 修改为 1，将 1 修改为 0。

你需要回答下面的问题 q 次（询问之间相互独立）：

给定集合 S ，先将编号在集合内的结点上的数字设为 1，其余设为 0。然后求至少需要几次操作才能使得整棵树上的数字都是 0。容易发现一定有解。

【输入格式】

从 `xor.in` 中读入数据。

第一行三个整数 sid, n, q ，其中 sid 表示测试点编号，它可以帮助你获得部分分。对于样例， sid 表示其对应的测试点编号。

接下来 $n - 1$ 个整数 f_2, f_3, \dots, f_n ， f_i 表示结点 i 的父亲。

接下来 q 行，每行表示一次询问：先输入 sz_i 表示 S 的大小，接下来 sz_i 个互不相等的整数表示集合内的元素。

【输出格式】

输出到 `xor.out` 中。

输出 q 行，每行一个整数表示答案。

【样例输入 1】

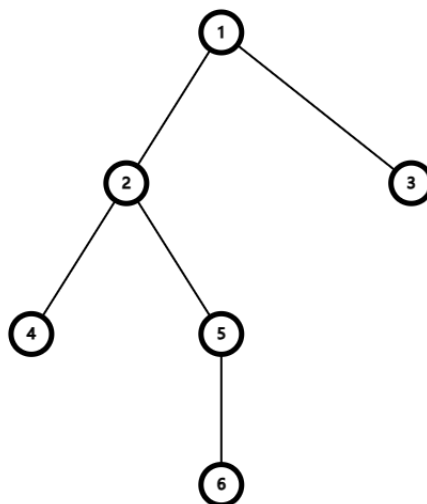
```
1 1 6 6
2 1 1 2 2 5
3 6 1 2 3 4 5 6
4 3 2 5 6
5 1 3
6 3 1 2 3
7 3 4 5 6
8 4 2 3 4 5
```

【样例输出 1】

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

【样例解释 1】

样例中的树如下：



一种最优策略如下：

- 第一次询问：对结点 1 进行操作。
- 第二次询问：依次对结点 2,4 进行操作。
- 第三次询问：对结点 3 进行操作。
- 第四次询问：依次对结点 1,4,5 进行操作。
- 第五次询问：依次对结点 4,5 进行操作。
- 第六次询问：依次对结点 2,3,6 进行操作。

让我们对第六次询问进行说明：

初始时结点 1 ~ 6 上面的数字分别为 0, 1, 1, 1, 1, 0。

对结点 2 进行操作后，结点 1 ~ 6 上面的数字分别为 0, 0, 1, 0, 0, 1。

对结点 3 进行操作后，结点 1 ~ 6 上面的数字分别为 0, 0, 0, 0, 0, 1。

对结点 6 进行操作后，结点 1 ~ 6 上面的数字分别为 0, 0, 0, 0, 0, 0。

操作次数不可能小于 3，所以答案是 3。

【样例 2】

见选手目录下的 *xor/xor2.in* 与 *xor/xor2.ans*。

【样例 3】

见选手目录下的 *xor/xor3.in* 与 *xor/xor3.ans*。

【样例 4】

见选手目录下的 *xor/xor4.in* 与 *xor/xor4.ans*。

【样例 5】

见选手目录下的 *xor/xor5.in* 与 *xor/xor5.ans*。

【样例 6】

见选手目录下的 *xor/xor6.in* 与 *xor/xor6.ans*。

【样例 7】

见选手目录下的 *xor/xor7.in* 与 *xor/xor7.ans*。

在上面的六个样例中，*sid* 都等于其对应的测试点编号。

【数据范围】

记 $\sum sz_i$ 为单个测试点内所有的 sz_i 的和。

对于所有测试数据, $1 \leq n, q \leq 2 \times 10^5$, $1 \leq sz_i \leq \min\{n, 10^5\}$, $\sum sz_i \leq 1.5 \times 10^6$ 。

测试点编号	$n, q \leq$	特殊性质
1 ~ 3	10	无
4 ~ 6	10^3	
7 ~ 8	2×10^5	A
9 ~ 11		B
12 ~ 14		C
15 ~ 16		D
17 ~ 20		无

- 特殊性质 A: 保证 $sz_i = 1$ 。
- 特殊性质 B: 保证集合中不存在两个数 x, y 使得树上 x 是 y 的父亲。
- 特殊性质 C: 保证 $f_i = i - 1$ 。
- 特殊性质 D: 保证 f_i 在 $[1, i - 1]$ 均匀随机生成。

【提示】

如果在运行大数据时出现 RE, 你可能需要开大栈空间。你可以在编译选项中加入如下指令来把栈空间开大到 512 MiB (请勿添加任何空格):

```
1 -Wl,-stack=536870912
```

本题目输入输出量大, 建议选手选用较快的输入输出方式。

如果你暂时不会解决整个问题, 请尝试解决特殊性质。

【后记】

小 N 向你保证, 如果你能 AC 这道题, 就不会再造有二十万个结点的圣诞树。

见字如面 (pair)

【题目背景】



送东阳马生序

明·宋濂

余幼时即嗜学。家贫，无从致书以观，每假借于藏书之家，手自笔录，计日以还。天大寒，砚冰坚，手指不可屈伸，弗之怠。录毕，走送之，不敢稍逾约。以是人多以书假余，余因得遍观群书。既加冠，益慕圣贤之道，又患无硕师、名人与游，尝趋百里外，从乡之先达执经叩问。先达德隆望尊，门人弟子填其室，未尝稍降辞色。余立侍左右，援疑质理，俯身倾耳以请；或遇其叱咄，色愈恭，礼愈至，不敢出一言以复；俟其欣悦，则又请焉。故余虽愚，卒获有所闻。

当余之从师也，负篋曳屣行深山巨谷中，穷冬烈风，大雪深数尺，足肤皲裂而不知。至舍，四支僵劲不能动，媵人持汤沃灌，以衾拥覆，久而乃和。寓逆旅，主人日再食，无鲜肥滋味之享。同舍生皆被绮绣，戴朱缨宝饰之帽，腰白玉之环，左佩刀，右备容臭，烨然若神人；余则缊袍敝衣处其间，略无慕艳意。以中有足乐者，不知口体之奉不若人也。盖余之勤且艰若此。今虽耄老，未有所成，犹幸预君子之列，而承天子之宠光，缀公卿之后，日侍坐备顾问，四海亦谬称其氏名，况才之过于余者乎？

今诸生学于太学，县官日有廩稍之供，父母岁有裘葛之遗，无冻馁之患矣；坐大厦之下而诵诗书，无奔走之劳矣；有司业、博士为之师，未有问而不告，求而不得者也；凡所宜有之书，皆集于此，不必若余之手录，假诸人而后见也。其业有不精，德有不成者，非天质之卑，则心不若余之专耳，岂他人之过哉！

东阳马生君则，在太学已二年，流辈甚称其贤。余朝京师，生以乡人子谒余，撰长书以为贽，辞甚畅达，与之论辨，言和而色夷。自谓少时用心于学甚劳，是可谓善学者矣！其将归见其亲也，余故道为学之难以告之。谓余勉乡人以学者，余之志也；诋我夸际遇之盛而骄乡人者，岂知予者哉？

小○望着窗外柳絮因风起，沉思良久，持笔，在信纸上郑重其事地写下：“宋濂君，见字如面！……”

【题目描述】

为了让宋濂感受到冬日的温暖，小O亲手织了 n 双手套，每双都是大小相同的两只，一只左手，一只右手。小O看着排列整齐的 $2n$ 只手套，并从左到右标上了 $1 \sim 2n$ ，很满意，打算装箱寄给宋濂。

但是，调皮的小O把这 $2n$ 只手套打乱了！

小O想把这些手套重新排成合法的排列。一个排列是合法的，当且仅当对于所有的 $i(1 \leq i \leq n)$ ，以下条件都成立：

- 位置 $2i - 1$ 和 $2i$ 上的手套大小相同。
- 位置 $2i - 1$ 的手套是左手手套。
- 位置 $2i$ 的手套是右手手套。

为了达成目标，小O可以进行一系列的操作。一次操作中，他选择相邻的两只手套进行交换位置的操作。两只手套相邻当且仅当两者编号差为 1。

小O犯难了，他织完手套就润了，所以你需要帮他求出：最少要做出多少次操作，才能得到一个合法的手套排列？

【输入格式】

从 *pair.in* 中读入数据。

第一行一个正整数 n ，表示有 n 双手套。

第二行 $2n$ 个整数 S_i ，第 i 个整数表示位置编号为 i 的手套。其中 $|S_u| \neq 0$ ，等于最初在位置 i 上的手套的大小。这里 $|x|$ 表示 x 的绝对值，当 $x \geq 0$ 时等于 x ，当 $x < 0$ 时等于 $-x$ 。如果 $S_i < 0$ ，则 i 位置上的手套是一只左手手套，否则是右手手套。

【输出格式】

输出到 *pair.out* 中。

输出一行一个整数，表示最少操作次数。

【样例输入 1】

```
1 2
2 2 1 -1 -2
```

【样例输出 1】

```
1 4
```

【样例输入 2】

```
1 3
2 -2 2 2 -2 -2 2
```

【样例输出 2】

```
1 1
```

【样例 3】

见选手目录下的 *pair/pair3.in* 与 *pair/pair3.ans*。

【样例 4】

见选手目录下的 *pair/pair4.in* 与 *pair/pair4.ans*。

【样例 5】

见选手目录下的 *pair/pair5.in* 与 *pair/pair5.ans*。

【数据范围】

本题采用捆绑测试。你能够得到一个子任务的分数，当且仅当你通过了这个子任务内的所有测试点。

对于所有数据：

- $1 \leq n \leq 10^5$ ；
- 对于所有 $i (1 \leq i \leq 2n)$ ，都有 $1 \leq |S_i| \leq n$ ；
- 总有某个合法的排列可以经由一系列对调而得到。

各个子任务的详细限制如下：

- 子任务 1 (10 分)： $n = 1$ 。
- 子任务 2 (20 分)： $n \leq 8$ 。
- 子任务 3 (20 分)：所有手套大小都是相同的。
- 子任务 4 (15 分)：所有在位置 $1, \dots, n$ 上的手套都是左手手套，而在位置 $n + 1, \dots, 2n$ 上的手套都是右手手套。而且对于所有 $i (1 \leq i \leq n)$ ，在位置 i 和 $i + n$ 上的手套大小相同。
- 子任务 5 (20 分)： $n \leq 10^3$ 。
- 子任务 6 (15 分)：无特殊限制。

【提示】

假如您做不出来或者 AK 了，您可以尝试背诵《送东阳马生序》的前两段，并默写下来发送到 yayay8888@outlook.com 这个邮箱中，小○也会为您送上冬日温暖！

加了个加 (number)

【题目背景】



【题目描述】

洛天依和乐正绫喜欢玩游戏。最近，她们喜欢上了一款叫做《加了个加》的游戏，这个游戏的规则如下：

乐正绫有一个整数 n ，洛天依有两个变量 a, b ，初始 $a = b = 0$ 。

洛天依可以进行如下四种操作（其中 \leftarrow 表示赋值）：

- 1: 令 $a \leftarrow a + 1$,
- 2: 令 $b \leftarrow b + 1$,
- 3: 令 $a \leftarrow a + b$,
- 4: 令 $b \leftarrow a + b$ 。

乐正绫想要让洛天依进行不超过 M 次操作，使得 $a = n$ ，而 b 可以为任意数。

洛天依感觉这个问题有点困难，于是找你帮忙。因为洛天依十分可爱，你需要帮助她完成这个任务。

【输入格式】

从 `number.in` 中读入数据。

一行两个整数 n, M 。

【输出格式】

输出到 *number.out* 中。

第一行一个整数 k ，表示使用的操作次数。你需要保证 $0 \leq k \leq M$ 。

接下来 k 行，每行一个整数 p ，表示这次使用的操作。你需要保证 $p \in \{1, 2, 3, 4\}$ 。

你需要保证每次运算过程中 $0 \leq a, b \leq 2 \times 10^{18}$ ，否则你会被判断为答案错误。

【样例输入 1】

```
1 5 200
```

【样例输出 1】

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

【样例解释 1】

共五次操作，每次操作后的 a, b 如下：

- 第一次： $a = 1, b = 0$,
- 第二次： $a = 1, b = 1$,
- 第三次： $a = 1, b = 2$,
- 第四次： $a = 3, b = 2$,
- 第五次： $a = 5, b = 2$ 。

显然是符合要求的，这仅仅是其中一种构造。

【样例 2】

见选手目录下的 *number/number2.in* 与 *number/number2.ans*。

这个样例，无疑是出题人最良心的馈赠。精心设计的 n 和 M ，富有提示性的构造方法，极高的数据强度，都表现了出题人极高的素质。出题人坚信，这个样例能够成为你 AK 道路上的有力援助。

【评分方式】

本题采用 Special Judge 评测你的程序，对于每个测试点，如果你的输出满足上述的所有要求，这个测试点得满分 4 分；否则得 0 分。

【数据范围】

对于所有测试数据： $1 \leq n \leq 10^{18}$ ； $200 \leq M \leq 2 \times 10^5$ 。

测试点编号	$n \leq$	$M \geq$	特殊性质
1, 2	10^2	200	无
3, 4, 5	10^3	200	
6, 7	10^4	1000	
8, 9		200	
10	10^7	3000	
11		200	
12	10^{12}	3000	
13		200	
14, 15, 16	10^{18}	3000	A
17, 18, 19		200	无
20, 21		3000	
22, 23, 24, 25		200	

特殊性质 A： n 是斐波那契数。形式化地，设 $f(1) = f(2) = 1, f(x) = f(x - 1) + f(x - 2) \ (x \geq 3)$ ，则 $\exists x \in \mathbb{N}^*, n = f(x)$ 。

【提示】

请自行检查你的输出是否合法。

过桥米线 (noodle)

【题目背景】



过桥米线已有一百多年的历史。相传，清朝时滇南蒙自市城外有一湖心小岛，一个秀才到岛上读书，秀才贤慧勤劳的娘子常常弄了他爱吃的米线送去给他当饭，但等出门到了岛上时，米线已不热了。

后来一次偶然送鸡汤的时候，秀才娘子发现鸡汤上覆盖着厚厚的那层鸡油有如锅盖一样，可以让汤保持温度，如果把佐料和米线等吃时再放，还能更加爽口。

于是她先把肥鸡、筒子骨等熟好清汤，上覆厚厚鸡油；米线在家烫好，而不少配料切得薄薄的到岛上后用滚油烫熟，之后加入米线，鲜香滑爽。

此法一经传开，人们纷纷仿效，因为到岛上要过一座桥，也为纪念这位贤妻，后世就把它叫做“过桥米线”。

在不知道哪里的 EotyhX 星上，也有这样一段关于“过桥米线”的传说。

【题目描述】

zdqhs6 河将 EotyhX 星的一个居民区分成了两块，分别是 A 区和 B 区。河的两岸都修了很多很多的高楼大厦，不多不少，两岸各 $10^9 + 7$ 栋。EotyhX 星人还对这些楼房进行了一个号的标，也就是说，两岸的建筑都从 0 标号一直到 $10^9 + 6$ 。A 区 i 号楼房恰好和 B 区 i 号楼房隔河相对。EotyhX 星人都有很严重的强迫症，他们让楼房两两间的距离恰好为 1，而 zdqhs6 河的宽度也恰好是 1。

故事要从送外卖说起。

这个居民区中有 n 个 EotyhX 星的外卖员，对于第 i 个外卖员，他需要在 A_i 区的 B_i 号楼房中取餐，并送往 C_i 区的 D_i 号楼房的顾客手中。

但是并不是所有外卖员的取餐和送餐点都在一个区域中，这个时候他就不得不开潜水艇去送外卖，**这会使外卖员一整天拉不出使！**

EotyhX 星政府为了增进民生福祉，于是决定在河上修桥。但是资金有限，所以只能修不超过 k 座桥。

还是因为强迫症，修的桥只能从 A 区的 i 号楼房到 B 区的 i 号楼房。

当政府修完桥后，外卖员就能够骑木马而不是乘坐潜艇去送餐啦！这个时候，送的食物就被称作——“过桥米线”。

还是为了增进民生福祉，政府希望在所有修桥的方案中，选出每位外卖员送餐距离和最小的方案。设第 i 位外卖员送餐距离为 dis_i ，求最小的 $\sum_{i=1}^n dis_i$ 。

【输入格式】

从 `noodle.in` 中读入数据。

输入的第一行包含两个正整数 k, n ，分别表示政府最多可以修的桥数和外卖员数量。

接下来 n 行，每一行包含四个参数： A_i, B_i, C_i, D_i ，表示对于第 i 个外卖员，他需要在 A_i 区的 B_i 号楼房中取餐，并送往 C_i 区的 D_i 号楼房的顾客手中。

【输出格式】

输出到 `noodle.out` 中。

输出仅为一行，包含一个整数，表示 $\sum_{i=1}^n dis_i$ 的最小值。

【样例输入 1】

```
1 1 5
2 B 0 A 4
3 B 1 B 3
4 A 5 B 7
5 B 2 A 6
6 B 1 A 7
```

【样例输出 1】

```
1 24
```

【样例输入 2】

```
1 2 5
2 B 0 A 4
3 B 1 B 3
4 A 5 B 7
5 B 2 A 6
6 B 1 A 7
```

【样例输出 2】

```
1 22
```

【样例输入 3】

```
1 2 1
2 A 0 A 0
```

【样例输出 3】

```
1 0
```

【数据范围】

本题采用捆绑测试。
所有数据都保证： $A_i, C_i \in \{A, B\}$ ； $0 \leq B_i, D_i \leq 10^9 + 6$ ； $1 \leq n \leq 10^5$ ； $1 \leq k \leq 2$ 。

子任务编号	$1 \leq n \leq$	$k =$	分值
1	样例	样例	0
2	10^3	1	10
3	10^5	1	20
4	10^2	2	10
5	10^3	2	20
6	10^5	2	40