

这大概是一套省选模拟题

AwD

2019.1

题目名称	沙雕出题	人与三	道良心题
源程序名	tree	sequence	N/A
输入文件名	tree.in	sequence.in	jigsaw.in
输出文件名	tree.out	sequence.out	jigsaw.out
单测试点时限	4s	4s	N/A
运行内存限制	512MB	512MB	N/A

哈，哈哈，哈哈哈，哈哈哈哈！

沙雕出题

steam 的年终促销刚刚结束，小 y^∞ 打开了自己的支付宝，掐指一算，震惊（并不）的发现，剩下的钱并不足以自己撑到月末。

“没办法，只能搞点钱来了。”

作为一个数竞选手，讲课是不可能讲课的，这辈子都不会讲课的，只能出出 OI 题，才能维持生活这个样子。

“……感觉，出不出题。”

没有枪、没有炮，敌人给我们造！

“首先，先搞一个原题。给一棵有点权的树。支持链加，求链上比 x 小的数的个数。”

“这个好像不太强，不如就问链上第 k 小吧。”

“都有链操作了，怎么能没有子树操作呢？还得支持子树加，求子树第 k 小。”

“说到子树操作，肯定树也得动起来哇，改父亲肯定是要有的吧”

“动态树肯定得强制在线的吧，没有强制在线怎么行呢？”

“……诶……这个，咋做啊……”

……

……

“能不能帮我验个题啊？”，说着，小 y^∞ 把他出的题发给了你。

“我不太验得动你的题啊……”

“……如果你验完了，而且我有闲钱的话，我可以给你点报酬哦……”

说到钱，就想起了化缘，今年下半年……好像不太对，不过反正你就答应下来了……

“那如果我做不出来，有没有题解供我参考呢？”

“啊，你是说题解是吧，我还没有编出……不对，我还没有编写完题解……”

“这样么，那我先试试看吧……”，说着，你打开了小 y^∞ 发给你的题面：

题目描述

给定一棵根为节点 1 的有根树，点上有点权。

支持以下五种操作：

- 对从节点 u 到根的链上的每个节点点权同时增加一个数。
- 对以节点 u 为根的子树中每个节点点权同时增加一个数。
- 修改节点 u 的父亲。
- 询问从节点 u 到根的链上的第 v 小的点权。
- 询问以节点 u 为根的子树中第 v 小的点权。

可能会要求强制在线。

输入格式

第一行一个整数，表示测试集编号。

第二行三个整数 N, M, K 。分别表示节点个数、操作次数及是否强制在线。

第三行 $N - 1$ 个整数，第 i 个表示节点 $i + 1$ 的父亲是 p ，保证 $p < i + 1$ 。

第四行 N 个整数，第 i 个为 V_i ，表示节点 i 的初始点权、

接下来 M 行，每行描述了一个操作。各行的开头有一个整数 opt ，用来描述操作的编号。

若 $opt = 1$ ，接下来两个整数 u, v 。表示对从节点 u 到根的链上的每个节点点权增加 v 。

若 $opt = 2$ ，接下来两个整数 u, v 。表示对以节点 u 为根的子树中每个节点点权增加 v 。

若 $opt = 3$ ，接下来两个整数 u, v 。表示将节点 u 的父亲修改为节点 v 。

若 $opt = 4$ ，接下来两个整数 u, v 。表示询问从节点 u 到根的链上的第 v 小的点权。

若 $opt = 5$ ，接下来两个整数 u, v 。表示询问以节点 u 为根的子树中第 v 小的点权。

如果 $K = 1$ ，则读入的 u, v 都需要异或上一次询问的答案 $lastans$ 。特殊的，当没有上一次询问时， $lastans = 0$ 。

输出格式

对于每个 $opt = 4$ 或 $opt = 5$ 的操作。输出一个整数表示答案。

样例

输入

```
0
5 5 0
1 1 2 2
1 2 3 4 5
1 2 1
2 2 2
3 4 2
4 2 1
5 2 2
```

输出

```
2
6
```

数据范围

一共 7 个测试集：

分数	测试集编号	N 的规模	M 的规模	K	特殊限制
9	1	$N \leq 5000$	$M \leq 5000$	$K = 0$	
11	2	$N \leq 50000$	$N \leq 50000$		i 的父亲恒为 $i - 1$
13	3				只存在 $opt = 1$ 或 $opt = 4$ 的操作
15	4				不存在 $opt = 3$ 的操作
17	5				不存在 $opt = 1$ 或 $opt = 4$ 的操作
10	6				
25	7			$K = 1$	

对于所有的数据，有： $1 \leq N, M$ ； $0 \leq V_i \leq 5 \times 10^8$ ；

对于 $opt = 1$ 或 $opt = 2$ 的操作，有 $0 \leq v \leq 10000$ 。

对于 $opt = 3$ 的操作，有 $v < u$ 。

对于 $opt = 4$ 或 $opt = 5$ 的操作，保证询问的点权一定存在。

人与三

宅文化分身 · 次世代亚文化之女 · 兽耳 JK · 全人类的希望 · 传奇驾驶员 · 可爱 (天下无双) · 一闪而过白色幻影 · 无情的粉丝屠杀机器 · 乱葬岗初代长老 · 月球魔术师 · 赫鲁晓夫亲女儿 · 承载着希望的独角兽 · 氦金发动机 · 一狐当关万夫暴毙 · super chat 收割机 · 大楚兴陈胜王 · 慈悲与全知全能的神 · 精神支柱 · AWSL · 大规模杀伤性武器 · 不凡的品茶之神 · Nippon 国宝 · 自闭雪是猫 · 百合营业鬼才 · 国家一级保护动物 · kawaii bass 公认第一音源 · 猫叫杀人吉尼斯世界纪录 · 工业化的高速神言 · 狐型自走 KTV 一式 · kitsune of Mass Destruction · 不败的神话 · 21c Renaissance · 共和国的伟大革命者 · C 酱, 是超自然部的一员, 也是小 y^∞ 的好朋友。

作为一个正统的超自然部的成员, C 酱的一大爱好是构筑魔法阵, 另一大爱好是屠杀粉丝 (确信)。众所周知, 在本位面, 绝大多数被称作魔法阵或与之相似的东西都不能产生超自然现象。但 C 酱坚信, 在自己摆出的魔法阵中, 总有一个是有效的, “谁知道世界有没有留后门呢?”。

作为共产主义的接班人, 小 y^∞ 不能接受 C 酱把大量时间花在这种虚无缥缈的事情上, 他想要用事实告诉 C 酱, 遍历所有的魔法阵是不现实的。经过日复一日的测量, 小 y^∞ 得到了 C 酱摆魔法阵的平均时间——11 分 45 秒 14。要算 C 酱尝试所有魔法阵的总时间, 就得算出有多少个她可能摆出的魔法阵了。

小 y^∞ 不喜欢“大约”、“估计”这类字眼, 他想知道个数的精确值。不过他并不擅长于数数, 因此只得求助于精通计算机的你。为了小 y^∞ 的幸福, 你会帮助他么?

题目描述

“C 酱的魔法阵遵循了一些规律, 除了 M 个圈以外, 其他部分长得都是一样的。”

“每个圈外都刻着祈祷文, 每则祈祷文都是某一单词的数次重复。”

“比如说我上次就见到了形如 AWSL AWSL AWSL 的祈祷文。”

“祈祷文当然不能太长咯。听 C 酱说, 祈祷文的长度之积不能超过 N 。”

“刚才说的那个祈祷文长度是 12, 如果有 3 个这样的圈, 那么长度之积就是 $12^3 = 1728$ 。”

“你问有多少不同的单词?”

“无穷无尽。不过, 对于一个特定的长度, 单词的数量是有限的, 它的数量还是一个关于长度的多项式——对于长度 x , 一共有 $\sum_{i=0}^K A_i x^i$ 种不同的单词。”

“插一发值就发现了啦 ~”

“圈本身就是是两两不同的。两个魔法阵不同, 当且仅当存在至少一个对应的圈。它们的单词不同, 或是单词出现的次数不同。”

“所以说, C 酱到底能画出多少个不同的魔法阵呢?”

.....
事实上, 上面的内容都不重要, 形式化的题意:

令 $H(x) = \sum_{i=0}^K A_i x^i$ 。

令 $G(x) = \sum_{d|x} H(d)$ 。

令 $F(x, y) = \begin{cases} G(x) & \text{if } y = 1, \\ \sum_{d|x} F(d, y-1)G(\frac{x}{d}) & \text{otherwise.} \end{cases}$ 。

求 $\sum_{i=1}^N F(i, M)$ 。

输入格式

第一行三个整数，分别为 N, M, K 。

第二行 $K + 1$ 个整数，第 i 个为 A_{i-1} 。

输出格式

一个整数表示答案，由于结果可能很大，请输出其在模 323232323 意义下的结果。

样例

输入

```
10 2 1
1 1
```

输出

```
753
```

数据范围

一共 9 个测试集：

分数	测试集编号	N 的规模	M 的规模	K 的规模
6	1	$N \leq 10^8$	$M \leq 1$	$K \leq 50$
12	2		$M \leq 2$	
6	3		$M \leq 30$	$K \leq 0$
12	4			$K \leq 1$
5	5	$N \leq 10^5$		$K \leq 50$
12	6	$N \leq 10^6$		
7	7	$N \leq 10^7$		
15	8	$N \leq 10^8$		
25	9	$N \leq 10^9$		

对于所有的数据，有： $1 \leq N, M$ ； $0 \leq K$ ； $0 \leq V_i < 323232323$ 。

道良心题

这是一道提交答案题

C 酱打开了小 y^∞ 的电脑。

C 酱输入了密码。

C 酱打开了“我的电脑”。

C 酱找到了一个被隐藏的文件夹。

C 酱很生气。

C 酱心生一计。

……三十分钟后……

C 酱关闭了小 y^∞ 的电脑。

题目描述

小 y^∞ 惊讶的发现，他电珍藏的插画被来自东方的神秘力量安排了！

经过仔细的观察，小 y^∞ 发现，这些图片先是被分割为了若干大小相等的矩形，而后位于中间的块被 shuffle 了。



原始图片



被打乱后的图片

由于一些知道的人知道，不知道的人不知道的原因，小 y^∞ 并不能访问这些插画的原图。于是乎，他只能尝试复原这些插画。可是小 y^∞ 是数竞选手，并不擅长编程，因此他只得求助于精通计算机的你，希望你能帮助他复原这些插画。

如果你成功的复原了这些插画的话，小 y^∞ 应该会把这些插画的出处告诉你的吧 ~

输入格式

下发文件中的 `in_image/X.ppm` 以 PPM 格式保存了编号为 X 的图片。PPM 格式是一种简单的图片存储格式，可以在 linux 下直接打开，且非常容易读取，是一种可以用来出 OI 题的好格式。

PPM 格式的结构如下：

第一行一个字符串，指明了编码的具体格式，在本题中，该字符串恒为 P6。

第二行两个整数 W, H ，分别为图片的宽与图片的高。

第三行一个整数 max_value ，表示图片最大的像素值，在本题中，该值恒为 255。

第四行一个长度为 $W \times H \times 3$ 的字符串，按先从上至下、再从左至右的顺序，每三个字符一组，依次描述了每个像素。字符的 ASCII 码依次对应了该像素的 RGB。

如果对读入感到困惑的话可以直接使用下发文件 `jigsaw_sample.cpp` 中的实现，调用

```
void read_image(const char *file_name, Image &img);
```

可以得到文件名为 `file_name` 的图片的信息，`img[i][j][k]` 是一个介于 0 至 1 之间的实数，描述了位于第 $i+1$ 行第 $j+1$ 列的像素的 "RGB"[k] 的值（即 0 为 R，1 为 G，2 为 B）。注：`img[0][0][0]` 为位于左上角的像素的红色的强度。

下发文件 `jigsaw.in` 描述了图片分块的信息，每行三个整数 X, h, w ，表示第 X 张图片中块的高是 h ，宽是 w 。也就是说，这张图片被分成了 $\frac{H}{h} \times \frac{W}{w}$ 块，其中有 $\frac{H-2h}{h} \times \frac{W-2w}{w}$ 块被打乱了。保证 H 是 h 的倍数， W 是 w 的倍数。

输出格式

由于图片太多，每张图片一个答案文件不现实，因此只需提交一个文件包含所有图片的答案。对于每张图片，输出若干行：

第一行一个整数 X ，表示这张图片的编号。

第二行两个整数，分别为 $\frac{H-2h}{h}$ 与 $\frac{W-2w}{w}$ 。（为什么要输出这两个东西？因为选手输出和标准答案格式一样才好看！）

接下来 $\frac{H-2h}{h}$ 行，每行 $\frac{W-2w}{w} \times 2$ 个整数，相邻两个数一组，若第 i 行的第 j 个数对为 k, p ，则表示被打乱图片中位于第 k 行的第 p 列的块在原始图片中位于第 i 行第 j 列。注：位于第 2 行的第 3 列的块的左上角为 $(2h, 3w)$ ，右下角为 $(3h-1, 4w-1)$ 。

具体实现可以参考下发文件 `jigsaw.out`，里面演示了应如何输出编号为 -10 至 -1 的这十张图片的答案。没有求出所有图片的答案？可以只输出求出来的答案。

样例

输入

-10 160 113

输出

-10
3 3
1 3 2 1 1 1
3 1 2 2 3 3
2 3 3 2 1 2

解释

这是下发文件中编号为 -10 的图片，即题面中的图片的答案。

评分方式

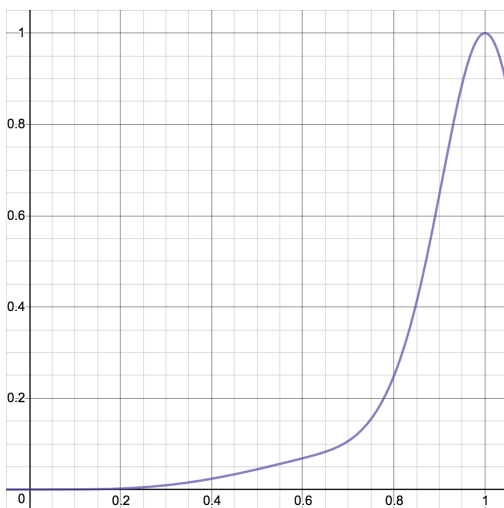
编号从 -10 至 -1 的图片为样例图片，提交的答案不计分。

对于编号为从 0 至 99 的每张图片，如果你拼对了 x 块，而一共需要拼的块有 y 块。则在你在该图片上的得分为：

$$p^3(1-p)\tanh(p)(2-\tanh(p)) + p\sin^{39}\left(\frac{\pi p}{2}\right) + \frac{1}{3}p(1-p)\sin^{47}\left(\frac{1.18\pi p}{2}\right)$$

其中 $p = \frac{x}{y}$ 。

这个函数的图像大约长这样：



评分函数

提示

如果对这个题感到迷茫的话，可以尝试直接在下发文件 `jigsaw_sample.cpp` 上修改。编译后运行 `./jigsaw_sample X`，程序会将编号为 X 的图片的答案写入 `jigsaw.out` 中，并将拼出的图片保存在 `out_image/X.ppm` 处。

如果想手玩这个题的话，我们提供了下发文件 `check.cpp` 来方便手玩。将编号为 X 的图片的答案手工写入 `jigsaw.out` 中，编译后运行 `./check X`，就可以在 `out_image/X.ppm` 处看到拼出的图片了！（真·良心出题人！）