

云南省青少年编程挑战精英赛

第四届 提高组

时间：2025 年 10 月 12 日 14:00 ~ 18:00

题目名称	「虚无」	「丰饶」偶遇	「智识」探测	「毁灭」预警
目录	ix	cross	detect	warning
可执行文件名	ix	cross	detect	warning
输入文件名	ix.in	cross.in	detect.in	warning.in
输出文件名	ix.out	cross.out	detect.out	warning.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
子任务数目	50	50	20	50
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	ix.cpp	cross.cpp	detect.cpp	warning.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. 选手提交的源程序请放在个人目录下题目英文名对应的子文件夹下；
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写；
3. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 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. 若无特殊说明，每道题的代码大小限制为 100KB；
7. 程序可使用的栈空间大小与该题内存空间限制一致；
8. 请务必使用题面中规定的编译参数，保证你的程序在评测机能够通过编译。此外不允许在程序中手动开启其他编译选项，一经发现，本题成绩以 0 分处理；
9. 禁止在代码中攻击交互库与评测系统，一经发现，本题成绩以 0 分处理；
10. 只提供 Windows 格式附加样例文件；
11. 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。

「虚无」(ix)

【题目描述】

作为「Lan」的令使，你不小心跌入了一个「虚无」的「空洞」。

你看见「空洞」内有 n 个神秘的圆柱体盒子，这些盒子中，每个盒子的底面半径和高度相等，不妨设第 i 个盒子高度为 h_i 。

通过一些尝试，你成功地把一个盒子挪到了另一个里面，因此你发现：若两个盒子 i, j 满足 $h_i < h_j$ ，你可以把盒子 i 挪到盒子 j 内部，从而封印住盒子 i ，使其不影响外界。

但可惜的是，一个盒子只能封印唯一一个盒子，不过如果一个盒子内部封印了另一个盒子，它仍然能被其它盒子封印，也就是说，盒子能嵌套合并。

形式化地说，假如第 j, k 个盒子都在第 i 个盒子内部，则必定有第 j 个盒子在第 k 个盒子内部或者第 k 个盒子在第 j 个盒子内部。

你观测到影响外界盒子数量与「空洞」消散的速度有关，具体而言，盒子越少，「空洞」就消散得越快。

你显然希望「空洞」消散得越快越好，因此为了尽快脱困，你想知道，经过一系列行动后，影响外界盒子数目最少为多少。

【输入格式】

从文件 **ix.in** 中读入数据。

输入的第一行包含一个正整数 n ，表示盒子的总数量。

输入的第二行包含 n 个正整数，第 i 个整数表示 h_i ，即盒子的高度。

【输出格式】

输出到文件 **ix.out** 中。

输出一行一个整数，表示影响外界的最少盒子数目。

【样例 1 输入】

```
1 8
2 5 4 5 2 4 4 2 3
```

【样例 1 输出】

```
1 3
```

【样例 1 解释】

用 $i(h)$ 表示第 i 个高度为 h 的盒子，以下为样例的一种嵌套情况：

1. $4(2) \rightarrow 8(3) \rightarrow 2(4) \rightarrow 1(5)$
2. $7(2) \rightarrow 5(4) \rightarrow 3(5)$
3. $6(4)$

此时能影响外界的盒子仅有 $4(2), 7(2), 6(4)$ 共 3 个，可以证明这是最小的答案。

【样例 2】

见选手目录下 `ix/ex_ix2.in` 与 `ix/ex_ix2.ans`。

该样例满足测试点 2 的条件。

【测试点约束】

对于所有测试点：保证 $1 \leq n \leq 100,000, 1 \leq h_i \leq 10,000$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$h_i \leq$	特殊限制
1 ~ 4	10	10,000	无
5 ~ 16	100,000	2	无
17 ~ 28	25	10,000	有
29 ~ 50	100,000	10,000	无

特殊性质：保证 h_i 在可能的范围内随机生成。

「丰饶」偶遇 (cross)

【题目描述】

作为「Lan」的令使，你在脱离「空洞」后看见了一只奇特的孽物。

这只孽物无法通过普通的攻击击败，因此你想要捕获它进行研究。可是「Lan」不擅活捉，所以祂给了你一支奇异的箭矢，它有一段锁定时间，只要你在锁定时间内使孽物下滑足够多的血量（有可能为负），这只箭矢就可以使它无法动弹，使你可以捕获之。

你仔细观察了孽物，发现它会使用一种持续回 R 点血量的技艺，而你有一种技能可以削减它的效果，并造成 C 的持续伤害，而你还有一把每次可以造成 X 点固定伤害的剑。

你还考量了自己的施放技能的水平，在锁定期间你最多可以施放 K 次技能。

具体而言，你在锁定期间有 n 轮的攻击时间，第 i 轮将会依次发生如下事情：

1. 孽物使用了 a_i 次回血技艺，形式化地说，孽物拥有一个效果栈，内有两个参数 r, c ，而这些技艺会使效果栈内的 r 增加 a_i ；
2. 你可以施放一次技能，这次技能将可以削减一层持续回血效果，并增加一层持续伤害效果，形式化地说，你的技能会使效果栈内的 r 减少 1（如果 $r = 0$ 则无法减少，但仍可施放技能）， c 增加 1，注意你最多施放 K 次技能；
3. 孽物回复 $r \times R$ 点血量；
4. 你使用你的剑造成 X 点固定伤害；
5. 持续伤害结算，孽物受到 $c \times C$ 点伤害。

你并不知道锁定期间你需要使孽物血量下降多少，所以你需要造成尽可能多的伤害，你需要输出这个最大下降值（有可能为负）。

注意：你并不关心锁定期间的血量最大下滑量，你只关心锁定结束时与锁定开始时孽物血量的差值。

【输入格式】

从文件 `cross.in` 中读入数据。

输入的第一行包含 5 个整数 n, K, R, X, C ，分别表示你的攻击轮次数、技能的最多施放次数、怪物的持续恢复血量值、每次攻击的固定伤害、技能的持续伤害值。

输入的第二行包含 n 个非负整数，第 i 个为 a_i ，表示第 i 轮怪物施放回血技艺的次数。

【输出格式】

输出到文件 `cross.out` 中。

输出一行一个整数，表示怪物下降血量的最大值。

【样例 1 输入】

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

【样例 1 输出】

```
1 43
```

【样例 1 解释】

最优方案下，你需要在第 1, 2, 3 轮使用技能，每回合怪物血量分别下滑：

1. $-0 \times R + X + 1 \times C = -0 \times 1 + 2 + 1 \times 3 = 5$ ，下面展示此轮 5 个事件的过程：

- (a) 孽物使用了 1 次回血技艺，增加一层回血效果，效果栈内 $r = 0 \rightarrow 1, c = 0$ ；
- (b) 你施放了一次技能，削减孽物一层回血效果（因为当前 $r = 1 > 0$ ），并使孽物增加一层持续伤害效果，效果栈内 $r = 1 \rightarrow 0, c = 0 \rightarrow 1$ ；
- (c) 孽物回复了 $r \times R = 0 \times 1 = 0$ 点血量；
- (d) 你用剑进行了一次攻击，造成固定 $X = 2$ 点伤害；
- (e) 持续伤害结算，孽物受到 $c \times C = 1 \times 3 = 3$ 点伤害；

2. $-0 \times R + X + 2 \times C = -0 \times 1 + 2 + 2 \times 3 = 8$ ；

3. $-1 \times R + X + 3 \times C = -1 \times 1 + 2 + 3 \times 3 = 10$ ；

4. $-1 \times R + X + 3 \times C = -1 \times 1 + 2 + 3 \times 3 = 10$ ；

5. $-1 \times R + X + 3 \times C = -1 \times 1 + 2 + 3 \times 3 = 10$ 。

总下滑 $5 + 8 + 10 + 10 + 10 = 43$ 点血量。

【样例 2 输入】

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

【样例 2 输出】

```
1 29
```

【样例 3 输入】

```
1 8 1 9 2 8
2 0 0 0 1 1 10 10 10
```

【样例 3 输出】

```
1 -520
```

【样例 4】

见选手目录下 *cross/ex_cross4.in* 与 *cross/ex_cross4.ans* 。
该样例满足测试点 2 的条件。

【样例 5】

见选手目录下 *cross/ex_cross5.in* 与 *cross/ex_cross5.ans* 。
该样例满足测试点 3 的条件。

【样例 6】

见选手目录下 *cross/ex_cross6.in* 与 *cross/ex_cross6.ans* 。
该样例满足测试点 4 的条件。

【样例 7】

见选手目录下 *cross/ex_cross7.in* 与 *cross/ex_cross7.ans* 。
该样例满足测试点 6 的条件。

【样例 8】

见选手目录下 *cross/ex_cross8.in* 与 *cross/ex_cross8.ans* 。
该样例满足测试点 15 的条件。

【样例 9】

见选手目录下 *cross/ex_cross9.in* 与 *cross/ex_cross9.ans* 。
该样例满足测试点 25 的条件。

【样例 10】

见选手目录下 *cross/ex_cross10.in* 与 *cross/ex_cross10.ans* 。
该样例满足测试点 27 的条件。

【样例 11】

见选手目录下 *cross/ex_cross11.in* 与 *cross/ex_cross11.ans* 。
该样例满足测试点 33 的条件。

【样例 12】

见选手目录下 *cross/ex_cross12.in* 与 *cross/ex_cross12.ans* 。
该样例满足测试点 35 的条件。

【样例 13】

见选手目录下 *cross/ex_cross13.in* 与 *cross/ex_cross13.ans* 。
该样例满足测试点 36 的条件。

【样例 14】

见选手目录下 *cross/ex_cross14.in* 与 *cross/ex_cross14.ans* 。
该样例满足测试点 38 的条件。

【样例 15】

见选手目录下 *cross/ex_cross15.in* 与 *cross/ex_cross15.ans* 。
该样例满足测试点 41 的条件。

【样例 16】

见选手目录下 *cross/ex_cross16.in* 与 *cross/ex_cross16.ans* 。
该样例满足测试点 45 的条件。

【样例 17】

见选手目录下 *cross/ex_cross17.in* 与 *cross/ex_cross17.ans* 。
该样例满足测试点 49 的条件。

【测试点约束】

对于所有测试点： $1 \leq K \leq n \leq 2,000,000, 0 \leq X, R, C, a_i \leq 1,000,000$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$K \leq$	$X, R, C, a_i \leq$	特殊限制
1	1	1	100	无
2	10	10	100	A
3	10	10	100	B
4, 5	10	10	100	C
6 ~ 10	10	10	100	无
11	300	300	1000	A
12	300	300	1000	B
13, 14	300	300	1000	C
15 ~ 20	300	300	1000	无
21	4000	4000	500,000	A
22	4000	4000	500,000	B
23, 24	4000	4000	500,000	C
25, 26	4000	50	500,000	无
27 ~ 32	4000	4000	500,000	无
33 ~ 34	2,000,000	10	100	无
35	2,000,000	10	500,000	无
36 ~ 37	2,000,000	2,000,000	100	A
38 ~ 40	2,000,000	2,000,000	100	B
41 ~ 44	2,000,000	2,000,000	100	C
45 ~ 48	2,000,000	2,000,000	100	无
49 ~ 50	2,000,000	2,000,000	1,000,000	无

特殊性质 A：保证 $R = 0$ 。

特殊性质 B：保证 $C = 0$ 。

特殊性质 C：保证 $a_i \in \{0, 1\}$ 。

「智识」探测 (detect)

【题目描述】

作为「Lan」的「令使」，你成功捕获了孽物。在对之的研究中，你发觉了一些不寻常的讯息，因此你决定布控一个探测网络，以监控一些不寻常的事件。

具体而言，你需要布置有 n 个测控点的网络，然后用以「家族缘结」制成的线相连。出于观测的需要，你只能用 $n - 1$ 条线使测控点与中心测控点直接或间接相连（即测控点与线构成了一棵树），每个测控点的观测数据经由若干条线和若干个测控点传递至中心观测点。

每个测控点有一个观测参数，为下列两个数据的和：与中心观测点距离（距离定义为两点间唯一简单路径上边的个数）、经由它传递的数据所属的观测点个数（即以中心观测点为根时子树内点数个数）。

测控点观测到的数据仅与整个城市的奇异事件，以及它的观测参数有关。因此你需要最大化观测网络的不同观测参数个数。

形式化的说，观测点网络为一棵有根树，每个点的观测参数为其子树内点的个数加上其与根的距离，你需要最大化整个树不同观测参数的个数。

请输出最大个数以及其中一种方案。

【输入格式】

从文件 *detect.in* 中读入数据。

输入的第一行包含一个整数 n ，表示测控点的个数。

【输出格式】

输出共 n 行，第一行包含两个整数 x, S ，分别表示最大的不同观测参数个数、你的方案的中心观测点。接下来 $n - 1$ 行每行两个整数 u, v ，表示观测网络的第 u, v 个观测点间有一条线。

注意：本题使用 Special Judge，为避免可能的不必要得分损失，请注意输出文件内每行尾不要有空格，文件末尾仅有一个换行。

本题已下发答案检查器。

选手目录下有文件 *detect/chk_user.cpp* 及相关库文件 *detect/testlib.h*。

请先使用命令 `g++ chk_user.cpp -o chk_user -std=c++14` 编译以得到答案检查器。

请在命令行环境下使用 `chk_user < 输入文件 > < 输出文件 > < 任意文件 >` 的格式调用检查答案器。

如果在使用答案检查器时遇到问题，请举手询问监考老师。

注意：评测环境下使用的答案检查器 `chk.cpp` 未下发，且不同于下发的答案检查器 `chk_user.cpp`，下发的答案检查器只会检查你的输出格式，以及你输出的图的答案是否为你声称的答案，故不保证通过答案检查器的答案一定正确。

注意：禁止任何攻击答案检查器的行为，违反此条的考生本题将作 0 分处理。

【样例输入】

1 2

【样例输出】

1 1 1

2 1 2

【样例解释】

此观测网络以 1 为中心观测点，对于每个点：

1. 与中心观测点的距离为 0，点 1, 2 的观测数据经过此点，观测参数为 $0 + 2 = 2$ ；
2. 与中心观测点的距离为 1，点 2 的数观测据经过此点，观测参数为 $1 + 1 = 2$ 。

不同的观测参数共有 1 种，可以证明，没有不同观测参数个数超过 1 的方案。

【测试点约束】

对于所有测试点： $1 \leq n \leq 1,000,000$ 。

每个测试点的具体限制见下表：

测试点编号	n
1	$= 1$
2	$= 2$
3	$= 3$
4	$= 4$
5	$= 5$
6	$= 6$
7 ~ 12	≤ 20
13 ~ 15	≤ 1000
13 ~ 20	$\leq 1,000,000$

「毁灭」预警 (warning)

【题目描述】

「Nous」向「Lan」发出预警,「Nanook」的「反物质军团」将不久发动对「Lan」治下城市的袭击,作为「Lan」的「令使」,你需要预测「反物质军团」可能造成的破坏。

将遭受袭击的城市区域共有 n 座城市,这些城市通过一些高速轨道相连,每条轨道都是双向的,连接两座不同的城市。「反物质军团」可以借助这些高速轨道入侵,为抵御袭击,你已经下令暂时切断一切不必要的轨道。因此,任意两座不同城市间,目前仅有一条由若干高速轨道组成的路径。

为躲避「Nous」可能发起的埋伏,「Nanook」的令使只打算让军团跃迁到一座城市开始发动袭击,虽然「Nanook」的军团非常强大,所有经过的城市都会被摧毁,但城市中也还有抵抗力量,因此军团不能经过城市且不摧毁此城市。此外,每条高速轨道只能承载能摧毁 k 座城市的军队通过。

在这些条件下,「Nanook」的「令使」将拟定一份作战计划,包含一座起始城市 and 所有预计摧毁的城市,然后他将在所有合法的计划中等可能的选取一个计划执行。

对于你而言,每个城市 i 都有一个重要程度 V_i ,你需要计算起始城市 and 所有被摧毁城市重要程度之和的期望。**注意:起始城市的重要程度会被计算两次。**

「Nous」在预警时给出了期望模 998,244,353 的值,你需要输出这个值以供校验。

形式化地说,如果你的答案为分数 $\frac{p}{q}$ 且 $\gcd(p, q) = 1$, 其中 $998,244,353 \nmid q$, 则必有唯一模意义下整数 x 满足 $xq \equiv p \pmod{998,244,353}$, 请输出 x 模 998,244,353 的值。

【输入格式】

从文件 `warning.in` 中读入数据。

本题的测试点包含有多组测试数据。

输入的第一行包含两个整数 c 和 t , 分别表示测试点编号和测试数据组数。对于样例, c 表示该样例与测试点 c 拥有相同的限制条件。

接下来,对于每组测试数据:

1. 第一行包含两个整数 n, k , 分别表示城市的总数、轨道能承载的军队量。
2. 接下来一行 n 个整数,第 i 个为 V_i , 表示第 i 个城市的重要程度。
3. 接下来 $n - 1$ 行,每行两个数字 u, v , 表示第 u 个城市和第 v 个城市间有一条未被切断的高速轨道。

【输出格式】

输出到文件 `warning.out` 中。

对于每组测试数据一行一个整数，表示答案对 998,244,353 取模的结果。

【样例 1 输入】

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

【样例 1 输出】

```
1 748683271
```

【样例 1 解释】

以 1 为起始点，1 是合法的方案，价值和为 $1 + 1 = 2$ （注意起始城市 1 的价值算了两次），12、13、123 都是合法的方案，价值和分别为 4、5、7，但 124 不是合法的方案，因为此时 12 这条边需要承载 2 个单位的军队，超过了 1 的承载限制，14 也不是合法的方案，因为 14 并不是一个连通块，军团不能不摧毁城市并经过此城市；

以 2 为起始点，合法的方案有 2、21、24、214，价值和分别为 4、5、8、9；

以 3 为起始点，合法的方案有 3、31，价值和分别为 6、7；

以 4 为起始点，合法的方案有 4、42，价值和分别为 8、10；

总共有 12 种方案，则期望为 $75 \div 12 = \frac{25}{4} \equiv 748683271 \pmod{998,244,353}$ 。

【样例 2 输入】

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

【样例 2 输出】

```
1 4
2 898419931
```

【样例 3 输入】

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

【样例 3 输出】

```
1 933543357
2 6
```

【样例 4】

见选手目录下 *warning/ex_warning4.in* 与 *warning/ex_warning4.ans* 。
该样例满足测试点 10 的条件。

【样例 5】

见选手目录下 *warning/ex_warning5.in* 与 *warning/ex_warning5.ans* 。
该样例满足测试点 12 的条件。

【样例 6】

见选手目录下 *warning/ex_warning6.in* 与 *warning/ex_warning6.ans* 。
该样例满足测试点 13 的条件。

【样例 7】

见选手目录下 *warning/ex_warning7.in* 与 *warning/ex_warning7.ans* 。
该样例满足测试点 27 的条件。

【样例 8】

见选手目录下 *warning/ex_warning8.in* 与 *warning/ex_warning8.ans* 。
该样例满足测试点 37 的条件。

【样例 9】

见选手目录下 *warning/ex_warning9.in* 与 *warning/ex_warning9.ans* 。
该样例满足测试点 41 的条件。

【样例 10】

见选手目录下 *warning/ex_warning10.in* 与 *warning/ex_warning10.ans* 。
该样例满足测试点 43 的条件。

【样例 11】

见选手目录下 *warning/ex_warning11.in* 与 *warning/ex_warning11.ans* 。
该样例满足测试点 44 的条件。

【样例 12】

见选手目录下 *warning/ex_warning12.in* 与 *warning/ex_warning12.ans* 。
该样例满足测试点 47 的条件。

【样例 13】

见选手目录下 *warning/ex_warning13.in* 与 *warning/ex_warning13.ans* 。
该样例满足测试点 49 的条件。

【测试点约束】

对于所有测试点： $1 \leq k \leq n \leq 8000, \sum n \leq 8000, 1 \leq V_i < 998,244,353$ ，保证方案数不被 998,244,353 整除。

每个测试点的具体限制见下表：

测试点编号	$\sum n \leq$	$k \leq$	特殊限制
1	1	1	无
2 ~ 5	10	10	无
6	10	10	A
7 ~ 9	10	10	B
10, 11	80	3	无
12	80	80	A
13 ~ 15	80	80	B
16 ~ 20	80	80	无
21, 22	250	3	无
23	250	250	A
24 ~ 26	250	250	B
27 ~ 30	250	250	无
31, 32	1500	3	无
33	1500	1500	A
34 ~ 36	1500	1500	B
37 ~ 40	1500	1500	无
41, 42	5000	3	无
43	5000	5000	A
44 ~ 45	5000	5000	B
46 ~ 48	5000	5000	无
49 ~ 50	8000	8000	无

特殊性质 A：保证高速轨道网络为一个菊花图，即存在一个城市（点），除其外每个城市（点）都与它有相连的高速轨道（边）。

特殊性质 B：保证高速轨道网络为一条链，即每个城市（点）的度数（与此城市相连的高速轨道数）不超过 2。