OIFC 2026 模拟赛

时间：2026 年 2 月 27 日 08:30 ∼ 13:30

题目名称	简单树上问题	时间旅行	序列变换 2
题目类型	传统型	传统型	传统型
目录	`tree`	`travel`	`change`
可执行文件名	`tree`	`travel`	`change`
输入文件名	`tree.in`	`travel.in`	`change.in`
输出文件名	`tree.out`	`travel.out`	`change.out`
每个测试点时限	5.0 秒	2.0 秒	3.0 秒
内存限制	512 MiB	256 MiB	1024 MiB
测试点数目	6	20	20
测试点是否等分	否	是	是

提交源程序文件名

对于 C++ 语言	`tree.cpp`	`travel.cpp`	`change.cpp`

编译选项

对于 C++ 语言	`-O2 -std=c++14 -static`

简单树上问题（tree）

【题目背景】

色彩本是无声的诗，藏在雾深处，落在光阴里，淡得几乎无痕。

蓝是天际遗落的凉，漫过云层，漫过山脊，像一场不肯醒来的清梦，安静，辽阔，又带着一点无人知晓的孤高。白是初雪未染的净，是月光铺在水面的薄霜，轻轻一碰，便散作满世界的清寂，不沾烟火，不惹尘埃。

青是远山含雾，是旧瓷微凉，是岁月沉淀下来的静。它不热烈，不张扬，只在风过林梢时，悄悄晕开一抹淡到极致的温柔，像心事被轻轻搁置。灰是云影掠过人间，不悲不喜，不浓不淡，将所有喧嚣都揉成朦胧，让万物在模糊里归于安宁。

最动人的颜色从不是盛放，而是淡去。

是光与影轻轻相拥，是暖与凉悄悄相融，

是繁华落尽后，留在眼底的那一片清宁。

世间声色皆可淡去，唯有这清冷的色彩，

如诗，如画，如一段无人打扰的时光，

轻轻落在心上，不喧哗，自有声。

【题目描述】

庭院里有一棵五彩缤纷的树。

树是由 n 个节点构成的简单无向无环图，每个节点具有它自己的颜色（用一个 $1 \sim n$ 的整数表示）。因为它是五彩缤纷的，所以它至少含有 5 种颜色。

太多的枝叶徒添冗杂，你只想要保留一个树上尽可能小的连通块，这个联通块依然要是五彩缤纷的（即它至少要有 5 种不同的颜色）。

问这个联通块最小由多少节点构成。

【输入格式】

第一行一个正整数 n 。

第二行包含 n 个正整数 a_i ，表示 i 号节点的颜色。

接下来 $n-1$ 行每行两个整数表示树上的一条边。

【输出格式】

一行一个整数表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
6
```

【数据范围】

子任务编号	$n \le$	$a_i \le$	分值
1	25	n	15
2	50	n	15
3	175	n	15
4	10^5	5	15
5	5×10^4	n	20
6	10^5	n	20

每个子任务有不超过 10 个测试点。

时间旅行（travel）

【题目描述】

6202 年，时间旅行的技术被发明出来了。然而，时间旅行的技术存在一定限制，只能在 m 个固定的节点之间进行，节点按照时间顺序依次编号为 $1 \sim m$。

现在有 n 名特工要执行任务，内容是从节点 s_i 出发，要将一种高危物品运送到节点 t_i。这种物品的运输容易带来时间线的混乱。具体而言，若这种物品在节点 x 开始进行时间旅行，在节点 y 结束，带来的混乱度为 $|x-y|$。

为了尽可能减少运输带来的混乱度，特工总局想出了一种方法：若两个特工同时到达了节点 y，他们可以在节点 y 暂时停留，这不会造成混乱度的增加。此时，他们可以交换所运输的物品。

现在，你需要求出最小的混乱度，并规划特工们的操作。具体方式如下：

- 一开始，所有特工携带物品开始时间旅行，特工 x 从 s_x 开始。
- 你可以进行操作 `0 x y` 表示让特工 x 在节点 y 暂停。要求特工 x 原来的位置不在 y，并且为了节约时间，要求特工 x 从原来位置向节点 y 移动的方向与从 s_x 向 t_x 移动的方向相同。
- 你可以进行操作 `1 x y` 表示让特工 x 与特工 y 交换物品。要求特工 x 与特工 y 当前在同一节点，且 $x \neq y$。
- 所有操作结束后，所有特工同时结束时间旅行，需要保证第 x 名特工在 t_x 站。你需要最小化此时的混乱度。

同时，为了减小出错的可能性，要求操作次数小于 4×10^5。

【输入格式】

第一行输入一个整数 T，表示数据组数。

对于每组数据：第一行两个正整数 n, m，含义如题面所示。

接下来 n 行，每行两个正整数，第 i 行的正整数分别表示 s_i, t_i。

【输出格式】

对于每组数据，第一行两个整数 ans, s，分别表示最小混乱度与操作总数。

接下来 s 行，每行三个整数 op, x, y 表示一次操作。

【样例 1 输入】

```
2
2 7
1 7
```

```
7 1
3 7
1 7
1 6
5 1
```

【样例 1 输出】

```
0 3
0 1 7
1 2 1
0 2 1
7 5
0 1 5
1 3 1
0 1 7
0 2 6
0 3 1
```

【样例 1 解释】

对于第一组数据，特工 1 移动到 7 号节点后与特工 2 交换了物品，然后特工 2 去到 1 号节点。

最终，特工 1 所持物品在 1 号节点开始时间旅行，在 1 号节点结束，带来的混乱度为 $|1-1|=0$，特工 2 所持物品带来的混乱度为 $|7-7|=0$，总混乱度为 0，可以证明这是混乱度的最小可能值。

【数据范围】

对于所有的数据，$1 \le T \le 10, 1 \le n \le 10^5, 1 \le m \le 10^6$。

测试点编号	n	m
1	$= 2$	$= 1000$
2	$= 100$	$= 2$
3	$= 4$	$= 5$
4	≤ 40	≤ 100
$5 \sim 6$	≤ 100	≤ 1000
$7 \sim 8$	≤ 2000	$\leq 2 \times 10^4$
9	$\leq 6 \times 10^4$	$\leq 10^5$
10	$\leq 10^5$	$\leq 10^6$

【提示】

为了方便选手测试，在附件中下发了 checker.cpp 文件，选手可以编译该程序，并使用它校验自己的输出文件。但请注意它与最终评测时所使用的校验器并不完全一致。你也不需要关心其代码的具体内容。

编译命令为：

```
g++ checker.cpp -o checker.exe -std=c++14
```

checker 的使用方式为：

```
./checker.exe <inputfile> <outputfile> <answerfile>
```

参数依次表示输入文件，你的输出文件与答案文件。

序列变换 2（change）

【题目描述】

你是一名公司老板。这个公司有 n 天的历史了，每天都有收益或者亏损。为了让账面好看，你不想让账本上出现亏损。你每次可以将一个收益的天的利润加到相邻的天上面。但你每做一次操作，被风险的风险就会变大。因此，你想要求出操作的最小次数。保证总收入是非负的。

形式化题意：给定一个长为 n 的序列 a_i，你每次可以选择一对相邻的 i 和 j，其中 $a_i > 0$，并让 $a_j = a_i + a_j, a_i = 0$，最后你要让所有 $a_i \geq 0$，求最小操作次数，保证 $\sum a_i \geq 0$。

【输入格式】

第一行一个整数 n。

第二行 n 个数，代表 a_i。

【输出格式】

一行一个整数，代表答案。

【样例 1 输入】

```
3
1 0 -1
```

【样例 1 输出】

```
2
```

【样例 2 输入】

```
5
-1 4 -1 -1 -1
```

【样例 2 输出】

```
5
```

【样例 1 输入】

```
6
-1 2 -1 -1 3 -1
```

【样例 3 输出】

```
5
```

【数据范围】

对于所有输入数据，满足 $|a_i| \le 10^3$。

测试点编号	$n \le$	特殊性质
$1 \sim 2$	10	
$3 \sim 4$	500	A
$5 \sim 8$	500	
$9 \sim 12$	2×10^3	
$13 \sim 20$	5×10^5	

特殊性质 A：$|a_i| = 1$。