

排序 (sort)

【问题描述】

牛牛最近学习了排序的相关知识，他对排序算法的运行效率产生了浓厚的兴趣并对此展开了一系列的研究。为了优化排序的运行时间，牛牛找到了来参加冬令营的你，希望你帮助他设计一台计算机进行排序。

这台计算机的设计方法如下：

- 需要进行排序的序列存储在大小为 n 的数组 Q 中。
- 进行计算的最小单元是比较器，一个比较器可以用一个三元组 (i, j, t) 表示，其中， i, j, t 均为正整数，其中 i, j 均在 $[1, n]$ 内且 $i < j$ 。它的功能是在时刻 t 比较 Q_i 与 Q_j 的大小，若 $Q_i > Q_j$ ，就交换 Q_i 与 Q_j 中的值，否则什么也不做。
- 若要让计算机正常运行，则需要任意两个比较器之间不能发生冲突。两个比较器 (A, B, T_1) 与 (C, D, T_2) 会发生冲突当且仅当 A, B, C, D 中有两个相等的数且 $T_1 = T_2$ 。例如：比较器 $(1, 3, 1)$ 和 $(2, 4, 1)$ 不会发生冲突，而比较器 $(1, 2, 3)$ 和比较器 $(2, 3, 3)$ 会发生冲突。

在运行时，这台计算机中的每个比较器都会按照预先设定好的参数在指定的时间进行相应的操作。这台计算机运行时需要消耗的时间为所有的比较器中参数 t 的最大值 M ，该值越小意味着运行时间越短。

牛牛发现，现实中的许多数据具有自己的特性。在设计计算机时往往需要考虑到这些特性，并针对性地进行设计才能达到好的效果。

为了检验你设计的计算机，牛牛提供了 8 个测试点共 100 组测试数据，每个测试数据包含 K 组测试数据，即 K 个长度为 n 的排列 P 。请你为每组测试数据设计一台运行时间不超过 m 的计算机，使得对于任意 $i \in [1, K]$ ，这台计算机能对 P_i 排序。

【输入格式】

输入文件 sort1.in~sort8.in 已在选手目录下。

每个输入文件第一行一个整数 $T (1 \leq T \leq 100)$ 表示这个测试点中的测试数据组数，同时也表示这个测试点的分值。在所有数据中， $\sum T = 100$ 。

每组数据第一行三个整数 $K, n, m (1 \leq n \leq 10^5, 1 \leq K, m \leq 10^4)$ ，意义在上文中已经说明。

接下来 K 行每行 n 个整数，描述一个 1 到 n 的排列。

数据保证有解。

【输出格式】

针对给定的8个输入文件sort1.in~sort8.in，你需要分别提交你的输出文件sort1.out~sort8.out。

对于每组测试数据，第一行输出一个整数 $num(1 \leq num \leq 10^6)$ 表示你设计的计算机中比较器个数。

接下来 num 行，每行 3 个整数 $u, v, t(1 \leq u < v \leq n, 1 \leq t \leq 150)$ ，描述你的计算机的一个比较器 (u, v, t) 。

【输入样例】

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

【输出样例】

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

【评分方式】

每个测试点单独评分，同时每个测试点你还有可能获得部分分。

每一个测试点总分为 T 分，每个测试点得分为其中所有测试数据的得分之和。

在每组测试数据中，若 $M \leq m$ ， $num \leq 10^6$ ，比较器之间不会发生冲突且可以对这 K 个排列正确地排序，则该测试点得 1 分，否则得 0 分。

如果你输出了不足 T 个计算机，那么我们将默认你输出的第 i 个计算机对应第 i 组测试数据。

如果你的输出不符合格式要求，我们不保证你能得到该测试点的分数。

【如何测试你的输出】

在终端中先切换到该试题的目录下。

```
cd sort
```

我们提供`checker`这个工具来测试你的输出文件是否是可接受的。使用这个工具的方法是，在终端中运行

```
./checker <input> <output>
```

其中<input> 是给出的输入文件，<output> 是你的输出文件，例如

```
./checker sort1.in sort1.out
```

将测试sort1.out 在以sort1.in 为输入时是否可以接受。

在你调用这个程序后，`checker` 将根据你给出的输出文件给出每一个测试数据的测试结果，其中包括（如果你输出的计算机同时出现了多种错误，将会返回其中一种）：

1. 非法退出：未知错误。
2. 输入文件错误：输入文件非法，在不修改输入文件的情况下不会触发。
3. The number of the comparators is invalid!: 输入的比较器个数不在 $[1, 10^6]$ 范围内，这时`checker` 将会直接退出。
4. Unexpected EOF: 输出文件中给出的计算机不完整。
5. The running time of the comparator should be in $[1, 150]$: 你给出的比较器的运行时间不在 $[1, 150]$ 范围内。
6. Invalid sorting network!: 排序网络不合法，包括 $u \geq v$ ， $u < 0$ ，比较器间产生冲突等。
7. Invalid! m=a but M=b: 计算机的运行时间超过限制。
8. The answer is incorrect: 排序网络合法，但是并没有将输入的排列正确排序。
9. Correct! m=a and M=b: 排序网络合法且对输入的排列正确排序，此时将得到该组测试数据的分数。
10. Total points: a: 如果`checker` 正常运行到了最后，将会额外输出一行表示你的总得分。其中 a 是你在这组数据中得到的分数。

【部分数据特性】

这里给出部分数据的部分特性：

1. 测试点 4 中，对于输入的每一个排列 P ，它每一个循环的大小都互不相同。循环可以理解为，将一个长度为 n 的排列 P 看成一张 n 个点 n 条边的无向图，其中 i 和 P_i 之间有一条边，这张图中的每一个联通块都是一个循环。
2. 测试点 7 中，对于输入的每一个排列 P ，都存在若干个互不相交的区间 $[l_i, r_i]$ ，使得在对每一个区间 $[l_i, r_i]$ 循环左移一次后，数组有序。循环左移的例子为：(1,2,3,4) 循环左移一步为 (2,3,4,1)。同时，该测试点的前 8 组数据，满足对于输入的每一个排列 P ，都存在整数 i ，使得对区间 $[1, i]$ 循环左移一次后，数组有序。

【提示】

请妥善保存输入文件 sort*.in 和你的输出文件 sort*.out，及时备份，以免误删。
通过自行修改输入文件而获得的得分是无效的。