

NOI 模拟赛

2025 年 6 月 20 日

题目名称	二重排列	IODS 模拟赛	染料商问题
题目类型	传统型	传统型	传统型
可执行文件名	dualperm	contest	dye
输入文件名	dualperm.in	contest.in	dye.in
输出文件名	dualperm.out	contest.out	dye.out
时间限制	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	20	捆绑测试	20

提交源程序文件名

对于 C++	dualperm.cpp	contest.cpp	dye.cpp
--------	--------------	-------------	---------

编译选项

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

注意事项（请仔细阅读）

1. 选手提交的源程序请放置在以题目对应文件名命名的子文件夹下。
2. 这套模拟赛的题目不一定按照难度顺序排列!!!
3. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
4. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
5. 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。
6. 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。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。

二重排列 (dualperm)

【题目描述】

小 P 有两个长度为 n 的 $(1, 2, \dots, n)$ 的排列 $A = (a_1, a_2, \dots, a_n), B = (b_1, b_2, \dots, b_n)$ 。同时, 小 P 发明了两种对排列的运算, 具体来说:

- 第 1 类运算: 对所有 $1 \leq i \leq n$ 计算 $a'_i = a_{b_{a_i}}$, 然后再对所有 $1 \leq i \leq n$ 赋值 $a_i = a'_i$ 。
- 第 2 类运算: 对所有 $1 \leq i \leq n$ 计算 $b'_i = b_{a_{b_i}}$, 然后再对所有 $1 \leq i \leq n$ 赋值 $b_i = b'_i$ 。

小 P 进行 q 次如上两类运算, 他想知道经过 q 次运算后, 排列 A, B 分别是什么。

【输入格式】

从文件 `dualperm.in` 中读入数据。

第一行输入一个整数表示 n 。

第二行输入 n 个整数表示排列 A 。

第三行输入 n 个整数表示排列 B 。

第四行输入一个整数表示 q 。

接下来 q 行, 第 i 输入一个整数 $type \in \{1, 2\}$ 表示第 i 次运算的类型。

【输出格式】

输出到文件 `dualperm.out` 中。

第一行输出 n 个整数表示所有运算过后的 A 。

第二行输出 n 个整数表示所有运算过后的 B 。

【样例 1 输入】

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

【样例 1 输出】

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

【样例 2】

见下发文件中的 *dualperm/ex_dualperm2.in* 与 *dualperm/ex_dualperm2.out*。

【测试点约束】

对于 30% 的数据，满足 $n, q \leq 5000$ 。

对于另外 20% 的数据，满足 $a_i = i (1 \leq i \leq n)$ 。

对于另外 20% 的数据，满足 $type = 1$ 。

对于全部数据，满足 $1 \leq n, q \leq 10^5$ 。

IODS 模拟赛 (contest)

【题目描述】

IODS 作为历史悠久的信息学竞赛赛事，有着极为丰富的模拟赛储备量。具体来说，IODS 一共有 n 场模拟赛，一场模拟赛恰好有 n 道题目，每道题目按照其难度从小到大划分为 1, 2, 3, 4, 5 这五个等级。对于 n 场模拟赛中的第 i ($1 \leq i \leq n$) 场，有 $a_{i,j}$ 表示这场模拟赛中难度为 j ($1 \leq j \leq 5$) 的题目数量，且满足 $\sum_{j=1}^5 a_{i,j} = n$ 。并且，每场模拟赛的题目都是按照难度顺序从小到大排列的。例如， $n = 6$ 且 $a_{i,1} = 1, a_{i,2} = 0, a_{i,3} = 2, a_{i,4} = 1, a_{i,5} = 2$ 时，模拟赛的题目顺序即为 [1, 3, 3, 4, 5, 5]。

对于任意一个 $(1, 2, \dots, n)$ 的排列 (p_1, p_2, \dots, p_n) ，小 P 可以用它和 IODS 的题库生成一组新的比赛：比赛的第 i 道题目为比赛 p_i 的第 i 道题目。而小 P 认为一场比赛是毒瘤的当且仅当一场比赛的题目顺序完全按照难度单调不升排列。小 P 想知道有多少个排列 P 能生成一场毒瘤的比赛，由于答案很大，你只需要求出答案对 2 取模的结果。

【输入格式】

从文件 `contest.in` 中读入数据。

第一行输入一个整数 n 。

接下来 n 行，第 i 行输入 5 个整数依次表示 $a_{i,1}, a_{i,2}, a_{i,3}, a_{i,4}, a_{i,5}$ 。对于每个 i ，当按题目难度自然升序排列时，各个 a_{ij} 对应的题目数正好就是该场比赛中的题目分布。

【输出格式】

输出到文件 `contest.out` 中。

输出一个整数，表示答案对 2 取模的结果。

【样例 1 输入】

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

【样例 1 输出】

```
0
```

【样例 2】

见下发文件中的 *contest/ex_contest2.in* 与 *contest/ex_contest2.out*。

【测试点约束】

本题采用捆绑测试，只有你通过了一个 subtask 中的所有数据才能获得其分数。

对于全部数据，满足 $1 \leq n \leq 2 \times 10^5$ ， $a_{i,j} \geq 0$ ， $\sum_{j=1}^5 a_{i,j} = n$ 。设 k 表示最大的 j 满足 $a_{i,j} \neq 0 (1 \leq i \leq n)$ 。

Subtask 1(10%)：满足 $k \leq 2$ 。

Subtask 2(20%)：满足 $k \leq 3$ ， $n \leq 500$ 。

Subtask 3(20%)：满足 $k \leq 3$ 。

Subtask 4(25%)：满足 $n \leq 500$ 。

Subtask 5(25%)：无额外限制。

染料商问题 (dye)

【题目描述】

G 国由 n 个城市组成，城市之间通过 $n - 1$ 条边互相连通，也就是说，这些城市构成了一个树的结构。

小 P 在公告面板上得知，有 m 个商贩在这 n 个城市中卖染料，且只有两种染料商：红染料商和蓝染料商。具体来说，面板上的第 i 条信息形如：有一个 R：红染料商或 B：蓝染料商在城市 p_i 卖染料。

在小 P 看来，如果一个染料商的分布局面是完美的，需要满足：对所有的红染料商 i ，要么存在一对不同的蓝染料商 j, k 使得城市 p_i 是城市 p_j 到城市 p_k 的唯一路径上的某一个点，要么存在一个蓝染料商 t 也位于城市 p_i ，即 $p_i = p_t$ 。

小 P 现在想知道如果只保留一个区间的染料商，即对于区间 $[l, r]$ ，仅保留 $l \leq i \leq r$ 的染料商 i ，哪些区间得到的分布局面是完美的。具体来说，你需要对 $1 \leq l \leq n$ 的每一个左端点 l ，求出最长的区间 $[l, r]$ 使其分布局面是完美的，你需要输出这个长度。

【输入格式】

从文件 `dye.in` 中读入数据。

第一行输入一个整数 n 。

接下来 $n - 1$ 行，每行输入两个整数 u, v 表示城市 u 和城市 v 之间有一条边。

接下来一行输入一个整数 m 。

接下来 m 行，每行输入一个字母 c_i 和整数 p_i ，若 c_i 为 'R' 则说明染料商 i 是红染料商，若 c_i 为 'B' 则说明染料商 i 是蓝染料商。

【输出格式】

输出到文件 `dye.out` 中。

输出 m 行，第 i 行一个整数表示 $l = i$ 时最大的长度。

【样例 1 输入】

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

```
5
B 5
B 5
B 5
B 5
R 5
```

【样例 1 输出】

```
5
4
3
2
0
```

【样例 2】

见下发文件中的 *dye/ex_dye2.in* 与 *dye/ex_dye2.ans*。

【样例 3】

见下发文件中的 *dye/ex_dye3.in* 与 *dye/ex_dye3.ans*。

【测试点约束】

对于 15% 的数据，满足 $n, m \leq 500$ 。

对于另外 15% 的数据，满足 c_i 为 'R' 的 i 的数量 $cnt_R \leq 2000$ 。

对于另外 10% 的数据，满足树的形态为一条链。

对于全部数据，满足 $1 \leq n, m \leq 10^5$ ， $1 \leq u, v \leq n$ ， $c_i \in \{R, B\}$ ， $1 \leq p_i \leq n$ 。保证给出的边构成一个 n 个点的树。