

# 2023 NOIP十连测-day3

题目名称	区间逆序对	排列	有向图最小环	二分图最大匹配
题目类型	传统型	传统型	传统型	传统型
英文题目名称	pair	arrange	ring	picture
输入文件名	pair.in	arrange.in	ring.in	picture.in
输出文件名	pair.out	arrange.out	ring.out	picture.out
每个测试点时限	2s	2s	1s	3s
内存限制	512MB	512MB	512MB	512MB
提交的源文件名	pair.cpp	arrange.cpp	ring.cpp	picture.cpp

【C++编译选项】 -lm -Wl,--stack=1000000000 -O2

【试题下载地址】 ftp://172.16.2.202/竞赛资料/20231004.zip 匿名访问即可

【结果上传地址】 ftp://172.16.2.202/20231004文件回收 用户名密码均为test

【赛后补题地址】 http://zhb.wms.edu/d/JH2023/ 训练

## [23NOIP十连测03]-T1-区间逆序对

### 【题目描述】

有一个长度为 $n$ 的序列 $a$ 。进行 $m$ 次询问，每组询问中给定 $l, r$ ，求有多少组 $i, j$ 满足 $l \leq i < j \leq r, a_i > a_j$ 。

输入格式

第一行，共两个整数，表示 $n, m$ 。

第二行，共 $n$ 个整数，表示 $a_1 \dots a_n$ 。

接下来 $m$ 行，每行两个整数，表示一组询问中的 $l, r$ 。

## 【输出格式】

共  $m$  行，每行一个整数，表示答案。

## 【输入样例】

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

## 【输出样例】

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

大样例见下发文件。

## 【数据范围】

对于 100% 的数据， $1 \leq n, m \leq 10^6, 1 \leq l \leq r \leq n, \forall i \in [1, n], 1 \leq a_i \leq 50$ 。

*Subtask1*(20%):  $n, m \leq 10^3$ 。

*Subtask2*(40%):  $n, m \leq 10^5$ 。

*Subtask3*(40%): 无特殊限制。

本题输入输出量较大，请使用较快的输入输出方式。

# [23NOIP十连测03]-T2-排列

## 【题目描述】

给定一个  $1 \sim n$  的排列  $a$ 。

有一个初始为空的序列  $b$ 。你可以进行任意多次操作，每次操作为以下三种之一：

- 设  $a'$  满足  $\forall i \in [1, n], a'_i = a_{(i \bmod n) + 1}$ ，将  $a$  替换为  $a'$ 。相当于将  $a$  向左循环移位一格。
- 设  $a'$  满足  $\forall i \in [1, n], a'_i = a_{((i + n - 2) \bmod n) + 1}$ ，将  $a$  替换为  $a'$ 。相当于将  $a$  向右循环移位一格。
- 选择一个  $i \in [1, m]$ ，并将  $a_i$  加入到  $b$  的末尾。

小 L 希望通过进行一些操作使得  $b$  的长度为  $n$  且从前往后依次为  $1 \sim n$ 。

你要求出最少需要进行多少次前两种操作。

## 【输入格式】

第一行，共两个整数，表示  $n, m$ 。

第二行，共  $n$  个整数，表示排列  $a$ 。

## 输出格式

共一行，一个整数，表示答案。

## 【输入样例】

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

## 【输出样例】

```
6
```

大样例见下发文件。

## 样例说明 1

一次操作一，此时  $a = (5, 1, 4, 2, 3)$ 。

一次操作三, 选择  $i = 2$ , 此时  $b = (1)$ 。

两次操作二, 此时  $a = (2, 3, 5, 1, 4)$ 。

两次操作三, 依次选择  $i = 1, i = 2$ , 此时  $b = (1, 2, 3)$ 。

一次操作二, 此时  $a = (4, 2, 3, 5, 1)$ 。

一次操作三, 选择  $i = 1$ , 此时  $b = (1, 2, 3, 4)$ 。

两次操作二, 此时  $a = (5, 1, 4, 2, 3)$ 。

一次操作三, 选择  $i = 1$ , 此时  $b = (1, 2, 3, 4, 5)$ 。

共使用 6 次前两种操作。

## 【数据范围】

对于 100% 的数据,  $1 \leq m \leq n \leq 5 \times 10^5$ 。

*Subtask1*(20%):  $n \leq 100$ 。

*Subtask2*(40%):  $n \leq 10^4$ 。

*Subtask3*(40%): 无特殊限制。

温馨提示: 某些 *STL* 真的非常非常慢, 使用前请谨慎考虑。

# [23NOIP十连测03]-T3-有向图最小环

## 【题目描述】

有一个有向图  $G$ 。  $\forall u, v \in [1, n], u \neq v$ , 如果  $a_{u,v} \neq 0$ , 那么  $G$  中存在一条  $u \rightarrow v$  的边权为  $a_{u,v}$  的边, 否则  $G$  中不存在  $u \rightarrow v$  的边。

小  $L$  想从点 1 沿着  $G$  中的边走到点  $n$ , 然后再走回点 1。

你要求出所有至少经过一次的边的边权和的最小值。保证有解。

## 【输入格式】

第一行, 共一个整数, 表示  $n$ 。

接下来  $n$  行，每行  $n$  个整数，第  $i$  行的第  $j$  个数表示  $a_{i,j}$ 。

当  $i = j$  时  $a_{i,j}$  没有意义，保证此时  $a_{i,j} = 0$ 。

## 【输出格式】

共一行，一个整数，表示答案。

## 【输入样例】

```
4
0 1 0 0
0 0 1 0
1 0 0 1
0 1 0 0
```

## 【输出样例】

5

大样例见下发文件。

## 【数据范围】

对于 100% 的数据， $1 \leq n \leq 200, \forall i, j \in [1, n], 0 \leq a_{i,j} \leq 10^9$ 。

*Subtask1*(30%) :  $n \leq 10$ 。

*Subtask2*(10%) :  $\forall i, j \in [1, n], a_{i,j} = a_{j,i}$ 。

*Subtask3*(40%) :  $n \leq 60$ 。

*Subtask4*(20%) : 无特殊限制。

数据巨水，觉得自己做法假了可以交原OJ。

# [23NOIP十连测03]-T4-二分图最大匹配

## 【题目描述】

有一个二分图  $G$ ，左右两边各有  $n$  个点，有不超过  $m$  条边，允许有重边。

左边的点标号为  $1 \sim n$ ，右边的点标号为  $n + 1 \sim 2n$ 。

设  $deg_i$  为点  $i$  的度数，即  $G$  中与  $i$  相连的边数。

定义  $match(G)$  表示  $G$  的最大匹配。

你需要对于每个  $k \in [1, n]$  求出  $match(G) = k$  时  $\sum_{i=1}^{2n} a_{i, deg_i}$  的最大值。

## 【输入格式】

第一行，共两个整数，表示  $n, m$ 。

接下来  $2n$  行，每行共  $m + 1$  个整数，第  $i$  行的第  $j$  个整数表示  $a_{i, j - 1}$ 。

## 【输出格式】

共一行， $n$  个整数，第  $i$  个数表示  $match(G) = i$  时的答案。

## 【输入样例】

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

## 【输出样例】

```
4 3
```

大样例见下发文件。

## 【数据范围】

对于 100% 的数据,  $1 \leq n \leq 30, 1 \leq m \leq 60, n \leq m, \forall i \in [1, 2n], j \in [0, m], 0 \leq a_i, j \leq 10^7$ 。

*Subtask1*(20%) :  $n \leq 4, m \leq 8$ 。

*Subtask2*(20%) :  $n \leq 8, m \leq 16$ 。

*Subtask3*(40%) :  $n \leq 20, m \leq 40$ 。

*Subtask4*(20%) : 无特殊限制。

文件格式要求：

```
-your_name
  |
  |--taskname1
  |       taskname1.cpp
  |
  |--taskname2
  |       taskname2.cpp
  |
  |--taskname3
  |       taskname3.cpp
  |
  |--taskname4
  |       taskname4.cpp
```