

# 20241122【NOIP】模拟测试

考试时间：2024年11月22日 7: 30 - 12: 00

题目	A.GCD=1的区间	B.心跳信竞俱乐 部	C.部队调动	D.双人成行
可执行文件名	gcd	doki	move	two
输入文件名	gcd.in	doki.in	move.in	two.in
输出文件名	gcd.out	doki.out	move.out	two.out
时间限制	2s	1s	2s	1s
空间限制	256MiB	256MiB	256MiB	256MiB
测试点数目	100	100	100	100
提交源程序文件名				
对于 C++ 语言	gcd.cpp	doki.cpp	move.cpp	two.cpp

## 提示

- 1. 函数main()的返回值类型必须是int，程序正常结束时的返回值必须是0。
- 2. 编译选项：-O2 -std=c++14 -static
- 3. 一定要使用 freopen 进行文件输入输出，文件名（程序名和输入输出文件名）必须使用英文小写。

```
1  freopen("文件名.in", "r", stdin);
2  freopen("文件名.out", "w", stdout);
```

- 4. 提交文件夹目录格式，样例如下：

```
1  /选手姓名
2      |-sample.cpp
3      |-sample.cpp
4      |-sample.cpp
5      |-sample.cpp
```

# A GCD=1的区间 (gcd)

2s / 256M

## 题目描述

给定一个由 1 到  $n$  组成的排列序列  $a$ ，以及  $Q$  组操作。每个操作会交换排列中两个位置上的元素。

对于每次交换操作后，你需要输出排列中子区间  $GCD$ （最大公约数）为 1 的子区间数量。

前面的操作会对后面的查询产生影响。

初始状态下的排列也需要输出一组结果。

具体地，设排列  $a = [a_1, a_2, \dots, a_n]$ ，定义子区间  $[l, r]$  的  $GCD$  为：

$$GCD(a[l:r]) = GCD(a_l, a_{l+1}, \dots, a_r)$$

其中  $GCD$  表示最大公约数。

## 输入格式

输入文件: gcd.in

第一行输入一个整数  $n$  表示排列的长度。

第二行输入一个长度为  $n$  的排列  $a$ ，其中  $a[i]$  表示第  $i$  个位置的元素。（ $1 \leq a[i] \leq n$ ，且元素不重复）

第三行输入一个整数  $Q$ ，表示操作的次数。

接下来  $Q$  行，每行包含两个整数  $x$  和  $y$ ，表示将排列中第  $x$  个位置和第  $y$  个位置的元素交换。

（ $1 \leq x, y \leq n$ ）

## 输出格式

输出文件: gcd.out

输出  $Q + 1$  行结果。第 1 行输出初始排列中子区间  $GCD$  为 1 的子区间数量。

接下来的  $Q$  行，每行输出一组交换操作后排列中子区间  $GCD$  为 1 的子区间数量。

## 输入样例

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

## 输出样例

1	7
2	6
3	7

### 样例说明

初始排列:  $[1, 2, 3, 4]$

子区间  $GCD$  为 1 的有:  $[1], [1, 2], [2, 3], [3, 4], [1, 2, 3], [2, 3, 4], [1, 2, 3, 4]$ , 共 7 个。

交换后: 交换位置 2 和 3, 排列变为  $[1, 3, 2, 4]$ 。

子区间  $GCD$  为 1 的有:  $[1], [1, 3], [3, 2], [1, 3, 2], [3, 2, 4], [1, 3, 2, 4]$ , 共 6 个。

再交换一次和初始排列一致

数据范围

	$n$ 范围	$Q$ 范围	特殊性质
20% 的数据	$1 \leq n \leq 100$	$1 \leq Q \leq 100$	
30% 的数据	$1 \leq n \leq 10^5$	$1 \leq Q \leq 10$	
30% 的数据	$1 \leq n \leq 10^5$	$1 \leq Q \leq 10^5$	排列和交换的位置是随机的
20% 的数据	$1 \leq n \leq 10^5$	$1 \leq Q \leq 10^5$	

# B 心跳信竞俱乐部 (doki)

1s / 256M

## 题目描述

心跳信竞俱乐部的 Monika 正在参加一场游戏，这场游戏的目标是得到等于或超过  $N$  的分数。

游戏的规则是这样的：

- 初始时，Monika 的总分  $X$  为 0，有一个  $M(M \geq 2)$  的骰子点数为  $1, 2, \dots, M$
- 每次开启一轮游戏，Monika 需要支付一枚金币。一轮游戏由若干个回合组成。在每个回合中，Monika 可以选择是否继续投掷骰子：
  - 如果 Monika 选择不投掷骰子，那么她把当前轮中“所有回合投掷的点数之和”的分数加给  $X$ ，然后该轮游戏结束。
  - 如果 Monika 选择投掷骰子：
    - 如果投掷结果为  $M$ ，则爆掉了，当前轮游戏立即结束，Monika 在当前轮拿不到任何分数。
    - 如果投掷结果不为  $M$ ，游戏继续进入下一回合。
- Monika 的目标是在花费最少金币的情况下，达到大于等于  $N$  的总分。

## 输入格式

输入文件: `doki.in`

输入仅包含两个整数  $N, M$ ：

- $N$  表示目标分数。
- $M$  表示骰子的一个特殊点数，若投掷出该点数，当前轮游戏将结束，且不会获得任何分数。

## 输出格式

输出文件: `doki.out`

输出一个实数，表示 Monika 在最优策略下，为了达到大于等于  $N$  的分数，所需的期望金币数，本题为  $SPJ$ ，要求  $1e - 5$  以内的相对误差或者绝对误差。

## 输入样例1

```
1 | 1 6
```

## 输出样例1

```
1 | 1.20000
```

### 样例1说明

只要扔出了非 6 的数字，立即结束当前轮。期望为 1.2 轮。

## 输入样例2

```
1 | 10 6
```

输出样例2

1 | 1.96438

输入样例3

1 | 1000 6

输出样例3

1 | 162.72836

数据范围

	<i>N</i> 范围	<i>M</i> 范围
20% 的数据	$1 \leq n \leq 10$	$2 \leq M \leq 6$
30% 的数据	$1 \leq n \leq 100$	$2 \leq M \leq 100$
50% 的数据	$1 \leq n \leq 1000$	$2 \leq M \leq 1000$

# C 部队调动 (move)

2s / 256M

## 题目描述

A 国有  $n$  座城市，这些城市之间由  $n$  条道路连接，形成了一张连通图。每条道路都是双向的，连接了两座城市，距离为 1。

在初始状态下，第  $i$  座城市拥有  $a[i]$  个士兵。为了应对新的军事部署计划，A 国需要重新安排士兵的分布，使得第  $i$  座城市需要达到  $b[i]$  个士兵。保证  $\sum_{i=1}^n a[i] = \sum_{i=1}^n b[i]$

为了确保部署计划能够顺利实施，所有士兵只能通过城市之间的道路移动。由于士兵移动成本高昂，因此需要计算出使得实际部署符合计划的最小移动总距离。

## 输入格式

输入文件: `move.in`

第一行输入一个整数  $n$ ，表示城市的数量。

接下来  $n$  行，每行包含两个整数  $u, v$ ，表示城市  $u$  和城市  $v$  之间有一条道路连接。 ( $1 \leq u, v \leq n$ )

输入保证图连通且没有重边。

接下来一行输入  $n$  个整数  $a[1], a[2], \dots, a[n]$ ，表示每个城市初始时拥有的士兵数量。 ( $0 \leq a[i] \leq 10^9$ )

最后一行输入  $n$  个整数  $b[1], b[2], \dots, b[n]$ ，表示每个城市最终需要的士兵数量。 ( $0 \leq b[i] \leq 10^9$ )

## 输出格式

输出文件: `move.out`

输出一个整数，表示实现部署所需的最小总移动距离。

## 输入样例1

```
1 | 4
2 | 1 2
3 | 2 3
4 | 3 4
5 | 4 1
6 | 4 0 0 0
7 | 0 1 1 2
```

## 输出样例1

```
1 | 5
```

### 样例1说明

- 2 个士兵：1 到 4
- 1 个士兵：1 到 2
- 1 个士兵：1 到 2, 2 到 3

总距离为 5

输入样例2

```
1 12
2 1 2
3 2 3
4 3 4
5 3 5
6 1 6
7 3 7
8 3 8
9 7 9
10 9 10
11 10 11
12 7 12
13 7 5
14 731210902 686861013 784003374 360957164 384909548
15 540545735 329842348 116445889 391794795 455551189
16 199747014 618522229
17 618522229 540545735 455551189 329842348 360957164
18 199747014 391794795 116445889 384909548 784003374
19 686861013 731210902
```

输出样例2

```
1 4632575453
```

输入样例3

```
1 5
2 4 2
3 2 5
4 5 1
5 1 3
6 3 4
7 1 2 2 3 0
8 2 2 0 1 3
```

输出样例3

```
1 7
```

数据范围

	<i>n</i> 范围	性质
20% 的数据	$1 \leq n \leq 100$	图是个环
30% 的数据	$1 \leq n \leq 10^5$	图是个环
50% 的数据	$1 \leq n \leq 10^5$	

# D 部队调动 (move)

1s / 256M

## 题目描述

在一个神秘的奇幻世界里，有一张由  $n$  个点和  $m$  条无向边组成的连通图。勇敢的 Alice 和聪明的 Bob 都在 1 号点，他们正准备接受一项极具挑战的任务。

在图上散落着  $k$  个任务，每个任务分别位于点  $p_1, p_2, \dots, p_k$ 。Alice 和 Bob 必须齐心协力完成这些任务，但他们的合作也有一些特别的规则：

- 1. 要完成第  $i$  个任务，Alice 或 Bob 中至少一人必须在节点  $p_i$ 。
- 2. 任务有严格的顺序：只有当 1 到  $i - 1$  的任务全部完成后，才能完成第  $i$  个任务。

Alice 和 Bob 想要在最短的时间内完成所有任务。在这个世界里，完成任务本身不需要时间，但走经过一条边，需要消耗 1 点时间。请你帮助他们计算完成所有任务所需的最短时间。（可以待在原地不动）

## 输入格式

输入文件: two.in

第一行包含三个整数  $n, m, k$ ，表示图的点数、边数以及任务的数量。

第二行包含  $k$  个整数  $p_1, p_2, \dots, p_k$ ，表示每个任务所在的位置。（ $1 \leq p_i \leq n$ ）

接下来  $m$  行，每行包含两个整数  $u, v$ ，表示一条连接点  $u$  和  $v$  的无向边。（ $1 \leq u, v \leq n$ ）

保证图连通。

## 输出格式

输出文件: two.out

输出一个整数，表示 Alice 和 Bob 完成所有任务的最短时间。

## 输入样例1

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

## 输出样例1

1	2
---	---

### 样例1说明

- 第一秒 Bob 从 1 到 2，立即完成任务 1，Alice 从 1 到 5
- 第二秒 Bob 从 2 到 3，立即完成任务 2，Alice 原地不动完成任务 3



输入样例2

1	40 50 8
2	3 8 10 12 24 25 26 38
3	1 2
4	1 3
5	1 4
6	1 8
7	1 9
8	2 5
9	2 21
10	2 22
11	2 39
12	3 7
13	3 25
14	3 31
15	4 6
16	4 17
17	4 25
18	4 30
19	5 11
20	5 13
21	5 35
22	6 10
23	6 22
24	7 18
25	7 36
26	8 15
27	8 26
28	8 29
29	8 35
30	10 14
31	10 20
32	11 12
33	11 16
34	11 31
35	11 33
36	13 28
37	13 40
38	14 27
39	16 19
40	17 21
41	17 27
42	19 23
43	19 37
44	20 40
45	22 24
46	23 29
47	23 36
48	24 38
49	25 26
50	25 36
51	27 32
52	29 34

输出样例2

## 数据范围

	$n$ 范围	$m$ 范围	$k$ 范围
10% 的数据	$1 \leq n \leq 10^4$	$0 \leq m \leq 10^4$	$1 \leq k \leq 2$
10% 的数据	$1 \leq n \leq 50$	$0 \leq m \leq 50$	$1 \leq k \leq 50$
30% 的数据	$1 \leq n \leq 150$	$0 \leq m \leq 150$	$1 \leq k \leq 50$
50% 的数据	$1 \leq n \leq 10^4$	$0 \leq m \leq 10^4$	$1 \leq k \leq 50$