

NOI 2025 模拟赛

第一试

| | | | |
|---------|----------|----------|-----------|
| 题目名称 | 排列 | 计算 | 查询 |
| 题目类型 | 传统型 | 传统型 | 传统型 |
| 目录 | perm | calc | query |
| 可执行文件名 | perm | calc | query |
| 输入文件名 | perm.in | calc.in | query.in |
| 输出文件名 | perm.out | calc.out | query.out |
| 每个测试点时限 | 1 秒 | 4 秒 | 2 秒 |
| 内存限制 | 512 MiB | 512 MiB | 512 MiB |
| 测试点数目 | 20 | 20 | 25 |
| 测试点是否等分 | 是 | 是 | 是 |

提交源程序文件名

| | | | |
|-----------|----------|----------|-----------|
| 对于 C++ 语言 | perm.cpp | calc.cpp | query.cpp |
|-----------|----------|----------|-----------|

编译选项

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

排列 (perm)

【题目描述】

给定一个 $1, 2, \dots, n$ 的排列 a_1, a_2, \dots, a_n , 保证 a 满足特殊性质 A 或特殊性质 B。你需要构造一个数列 b_1, b_2, \dots, b_L , 使得:

- $n \leq L \leq 5.5 \times 10^5$ 。
- 对于任意 $1 \leq i \leq L$, $1 \leq b_i \leq n$ 。
- 对于任意 $1 \leq i \leq n$, $b_i = i$, 且 $b_{L-n+i} = a_i$ 。
- 对于任意 $1 \leq i < j \leq L$, 如果 $j - i \leq n - 2$, 则 $b_i \neq b_j$ 。

如果有多个满足条件的数列 b , 输出任意一个均可。可以证明至少有一个满足条件的数列 b 。

【输入格式】

从文件 `perm.in` 中读入数据。

第一行, 一个正整数 n 。

第二行, n 个正整数 a_1, a_2, \dots, a_n 。

【输出格式】

输出到文件 `perm.out` 中。

第一行, 一个正整数 L 。

第二行, L 个正整数 b_1, b_2, \dots, b_L 。

【样例 1 输入】

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

【样例 1 输出】

```
1 14
2 1 2 3 4 2 1 4 2 3 1 2 4 1 3
```

【样例 2 输入】

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

【样例 2 输出】

```
1 26
2 1 2 3 4 5 6 1 2 3 4 6 1 5 2 3 4 6 1 5 2 4 3 1 6 2 5
```

【数据范围】

对于所有数据, $3 \leq n \leq 1000$ 。

| 测试点 | $n \leq$ | 特殊性质 |
|---------|----------|------|
| 1 ~ 4 | 10 | A |
| 5, 6 | 20 | |
| 7, 8 | 100 | |
| 9, 10 | 500 | |
| 11 ~ 14 | 1000 | B |
| 15 ~ 20 | | A |

特殊性质 A: a 在所有 $1, 2, \dots, n$ 的排列中随机生成。

特殊性质 B: 保证恰好有两个位置 k 满足 $a_k \neq k$ 。

计算 (calc)

【题目描述】

给定正整数 n , 有两个长度为 n 的整数数列 $\{a_n\}, \{b_n\}$, 请你找到两个非负整数 x, y 和一个非负整数数列 $\{c_n\}$, 最小化

$$\sum_{i=1}^n (|a_i - c_i x| + |b_i - c_i y|).$$

你只需要输出这个式子的最小值即可。

【输入格式】

从文件 *calc.in* 中读入数据。

第一行, 一个正整数 n 。

第二行, n 个非负整数 a_1, a_2, \dots, a_n 。

第三行, n 个非负整数 b_1, b_2, \dots, b_n 。

【输出格式】

输出到文件 *calc.out* 中。

一行, 一个非负整数, 表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 4
```

【样例 2】

见选手目录下的 *calc/calc2.in* 与 *calc/calc2.ans*。

【样例 3】

见选手目录下的 *calc/calc3.in* 与 *calc/calc3.ans*。

【样例 4】

见选手目录下的 *calc/calc4.in* 与 *calc/calc4.ans*。

【样例 5】

见选手目录下的 *calc/calc5.in* 与 *calc/calc5.ans*。

【子任务】

对于所有数据， $1 \leq n \leq 30$ ， $0 \leq a_i, b_i \leq 10^8$ 。

| 测试点 | $n \leq$ | 特殊性质 |
|---------|----------|----------------------|
| 1 ~ 3 | 30 | $a_i, b_i \leq 100$ |
| 4 ~ 7 | | $a_i, b_i \leq 10^5$ |
| 8 | | $b_i = 0$ |
| 9 ~ 14 | 8 | 无 |
| 15 ~ 17 | 25 | |
| 18 ~ 20 | 30 | |

查询 (query)

【题目描述】

给定长度为 n 的数列 c_1, c_2, \dots, c_n , 保证 c 中的元素互不相同。

有一个长为 n 的链, 其中点 i 与点 $i+1$ 间的无向边的边权为 d_i 。

对于一个区间 $[l, r]$, 设 k 是 c_l, c_{l+1}, \dots, c_r 中最小值的下标, 令 D 是在链上从点 k 开始, 同时经过点 l 和点 r , 最终停在点 l 或点 r 的路径的边权和的最小值, 定义 $[l, r]$ 的权值是 $c_k \cdot D$ 。

有 q 次查询, 每次给定区间 $[L, R]$, 请求出所有满足 $[l, r] \subseteq [L, R]$ 的区间 $[l, r]$ 的权值最大值。

【输入格式】

从文件 **query.in** 中读入数据。

第一行, 一个正整数 n 。

第二行, n 个正整数 c_1, c_2, \dots, c_n 。

第三行, $(n-1)$ 个正整数 d_1, d_2, \dots, d_{n-1} 。

第四行, 一个正整数 q 。

接下来 q 行, 每行两个正整数 L, R , 表示一次询问。

【输出格式】

输出到文件 **query.out** 中。

q 行, 每行一个正整数, 表示答案。

【样例 1 输入】

```
1 5
2 4 1 3 2 5
3 1 2 3 4
4 6
5 1 1
6 2 4
7 2 5
8 1 5
9 1 3
10 1 4
```

【样例 1 输出】

```
1 0
2 6
3 20
4 20
5 4
6 7
```

【样例 2】

见选手目录下的 *query/query2.in* 与 *query/query2.ans*。

【样例 3】

见选手目录下的 *query/query3.in* 与 *query/query3.ans*。

【样例 4】

见选手目录下的 *query/query4.in* 与 *query/query4.ans*。

【样例 5】

见选手目录下的 *query/query5.in* 与 *query/query5.ans*。

【样例 6】

见选手目录下的 *query/query6.in* 与 *query/query6.ans*。

【样例 7】

见选手目录下的 *query/query7.in* 与 *query/query7.ans*。

【子任务】

对于所有数据, $1 \leq n \leq 2 \times 10^5$, $1 \leq q \leq 5 \times 10^5$, $1 \leq c_i \leq 10^8$, $1 \leq d_i \leq 10^5$, $1 \leq L \leq R \leq n$ 。

| 测试点 | $n \leq$ | $q \leq$ | 特殊性质 |
|---------|-----------------|-----------------|------|
| 1 ~ 3 | 5000 | 5×10^5 | 无 |
| 4 ~ 6 | 2×10^5 | 100 | |
| 7 ~ 12 | | 5×10^5 | A |
| 13 ~ 16 | | | B |
| 17 ~ 20 | 10^5 | 2×10^5 | 无 |
| 21 ~ 25 | 2×10^5 | 5×10^5 | |

特殊性质 A：保证数列 c 单调增。

特殊性质 B：保证 c_i, d_i 随机生成。