

信号连接 (signal)

【题目描述】

太空中有 n 个通信站，编号为 $1 \sim n$ 。每个通信站均有两个参数，其中通信站 i ($1 \leq i \leq n$) 的发射系数为 a_i ，接收系数为 b_i 。

若通信站 i ($1 \leq i \leq n$) 向通信站 j ($1 \leq j \leq n$) 发射信号，则这两个通信站间会建立起双向信号连接，所需代价为 $a_i b_j$ 。由于通信站可能吸收宇宙射线中的能量，因此所需代价可能为负数。若要在通信站 i, j ($1 \leq i, j \leq n$) 间建立双向信号连接，可以任选其中一个通信站作为发射方发起连接，故最小代价为 $\min(a_i b_j, a_j b_i)$ 。当两个通信站间直接建立了信号连接，或通过其他通信站间接建立了信号连接，这两个通信站便可以加入同一个通信系统中。

为了减少成本，往往只有一部分的通信站处于运作状态。你要求出，对于编号处于区间 $l \sim r$ 中的通信站，当只有这些通信站间可以建立信号连接时，建立尽可能少的信号双向连接将这些通信站全部加入同一个通信系统的最小代价。

【实现细节】

选手不需要，也不应该实现 `main` 函数。

选手需要确保提交的程序包含头文件 `signal.h`，即在程序开头加入以下代码：

```
1 #include "signal.h"
```

选手需要在提交的程序源文件中实现以下两个函数：

```
1 void signal(int n, std::vector<int> a, std::vector<int> b);
```

- n 表示通信站的个数。
- 对于 $1 \leq i \leq n$ ， a_i 表示通信站 i 的发射系数， b_i 表示通信站 i 的接收系数。注意： a, b 是两个长度为 $n + 1$ 的序列，其中 a_0, b_0 无实际含义。
- 注意：对于每个测试点，该函数可能会被交互库调用多次。

```
1 long long mincost(int l, int r);
```

- l, r 表示处于运作状态的通信站的编号区间。
- 该函数需要返回建立尽可能少的信号双向连接将这些通信站全部加入同一个通信系统的最小代价。
- 注意：通信站的发射系数与接收系数为最近一次调用 `signal` 函数时传入的参数。
- 所有该函数的调用均在调用至少一次 `signal` 函数之后。

注意：在任何情况下，交互库运行所需时间均不会超过 0.1 秒，所用内存为固定大小，且均不超过 64 MiB。

【测试程序方式】

试题目录下的 `grader.cpp` 是交互库参考实现，最终测试时所用的交互库实现与该参考实现有所不同，因此选手的解法不应该依赖交互库实现。

选手可以在本题目录下使用如下命令编译得到可执行程序：

```
1 g++ grader.cpp signal.cpp -o signal -std=gnu++14 -O2 -pipe -static -s
```

对于编译得到的可执行程序：

- 可执行文件将从**标准输入**读入以下格式的数据：
 - 输入的第一行包含一个正整数 t ，表示测试数据组数。
 - 接下来依次为每组测试数据，对于每组测试数据：
 - * 第一行包含两个正整数 n, q ，分别表示通信站的个数与询问的次数。
 - * 第二行包含 n 个整数 a_1, \dots, a_n ，分别表示每个通信站的发射系数。
 - * 第三行包含 n 个整数 b_1, \dots, b_n ，分别表示每个通信站的接收系数。
 - * 第 $i+3$ ($1 \leq i \leq q$) 行包含两个正整数 l, r ，表示第 i 次询问时处于运作状态的通信站的编号区间。
- 可执行文件将输出以下格式的数据至**标准输出**：
 - 对于每组测试数据，输出一行 q 个整数，分别表示每次询问时建立尽可能少的信号双向连接将处于运作状态的通信站全部加入同一个通信系统的最小代价。

【样例 1 输入】

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

【样例 1 输出】

```

1 5
2 -33 -47

```

【样例 1 解释】

该样例共包含两组测试数据。

对于第一组测试数据，询问时所有的通信站均处于运作状态，可以从通信站 1 向通信站 2,3 发射信号，所需代价为 $1 \times 1 + 1 \times 4 = 5$ 。此时所有通信站可以全部加入同一个通信系统。

【附加文件说明】

在附加文件中：

1. `grader.cpp` 是提供的交互库参考实现。
2. `signal.h` 是头文件，选手不用关心具体内容。
3. `template_signal.cpp` 是提供的示例代码，选手可参考并实现自己的代码。

【子任务】

设 N, Q 分别表示单个测试点中所有测试数据的 n, q 的和。对于所有测试数据，均有：

- $1 \leq t \leq 10^5$;
- $1 \leq n, q \leq 10^5, N, Q \leq 10^5$;
- 对于所有 $1 \leq i \leq n$ ，均有 $|a_i|, |b_i| \leq 10^6$;
- $1 \leq l \leq r \leq n$ 。

子任务编号	分值	$N, Q \leq$	特殊性质
1	3	10^2	无
2	7	5,000	A
3	11		B
4	13		无
5	14	10^5	A
6	16		B
7	36		无

特殊性质 A：对于所有 $1 \leq i \leq n$ ，均有 $a_i, b_i > 0$ 。

特殊性质 B：对于所有 $1 \leq i \leq n$ ，均有 $a_i b_i < 0$ 。

【评分方式】**注意：**

- 选手不应当通过非法方式获取交互库的内部信息，如直接与标准输入、输出流进行交互。此类行为将被视为作弊；
- **最终的评测交互库与样例交互库的实现不同。**

本题首先会受到和传统题相同的限制，例如编译错误会导致整道题目得 0 分，运行时错误、超过时间限制、超过空间限制等会导致相应测试点得 0 分等。选手只能在程序中访问自己定义的变量以及交互库给出的变量，尝试访问其他地址空间将可能导致编译错误或运行错误。

在上述条件基础上：

- 对于每个测试点，程序得到满分当且仅当每次调用 `mincost` 函数时返回的答案均正确。