

## Mục lục

<b>Contest #4 – HARD LEVEL</b>	<b>2</b>
Bảng hoán vị — PERTAB . . . . .	2
COMMUTER . . . . .	4
Lầu cô đơn . . . . .	6
<b>Contest #3 – MEDIUM LEVEL</b>	<b>8</b>
Bầu cán bộ lớp — CANBO . . . . .	8
Thay đổi gốc – REROOT . . . . .	9
MAZETRAP . . . . .	11
<b>Contest #2 – HARD LEVEL</b>	<b>13</b>
Biểu quyết – OPINION . . . . .	13
GARAGE . . . . .	14
Thủ thư . . . . .	16
<b>Contest #1 – MEDIUM LEVEL</b>	<b>18</b>
Đỉnh xanh đỏ – RBVER . . . . .	18
Hợp nhất đoạn – MERGING . . . . .	19
Hoán đổi . . . . .	21

Nộp bài tại: [dalatcamp.contest.codeforces.com](https://dalatcamp.contest.codeforces.com)

# Bài 1. Bảng hoán vị — PERTAB

File dữ liệu vào: PERTAB.INP

File kết quả: PERTAB.OUT

Alice vừa học về một vài kỹ thuật bảng số trong môn lập trình thuật toán. Buổi trước Alice cũng đã được học về các dãy số hoán vị. Khá thích thú với các con số trên bảng số cũng như tính chất của hoán vị, hôm nay Alice muốn chơi một trò chơi mới lạ liên quan đến cả hai vấn đề này.

Ban đầu, Alice thiết lập một bảng số  $N \times N$  gồm  $N$  hàng được đánh số từ 1 đến  $N$  từ trên xuống dưới và gồm  $N$  cột được đánh số từ 1 đến  $N$  từ trái sang phải. Hàng  $i$  và cột  $j$  là ô  $(i, j)$  được Alice điền giá trị  $a_{i,j}$ , và sau cùng thì bảng số thỏa mãn tính chất: *mỗi hàng và mỗi cột là một hoán vị độ dài  $N$* .

Ví dụ bảng  $\begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{pmatrix}$  là bảng  $3 \times 3$  thỏa mãn tính chất trên. Những bảng số thỏa mãn tính chất này

Alice gọi là *bảng hoán vị*.

Tiếp theo Alice thực hiện các loại thao tác sau, mỗi loại tương ứng với một kí tự:

- $R$  : dịch chuyển các cột sang bên phải, tức là thay đổi giá trị  $a_{i,j}$  thành  $a_{i,(j-2) \bmod N + 1}$ ;
- $L$  : dịch chuyển các cột sang bên trái, tức là thay đổi giá trị  $a_{i,j}$  thành  $a_{i,(j \bmod N) + 1}$ ;
- $D$  : dịch chuyển các hàng xuống dưới, tức là thay đổi giá trị  $a_{i,j}$  thành  $a_{((i-2) \bmod N) + 1, j}$ ;
- $U$  : dịch chuyển các hàng lên trên, tức là thay đổi giá trị  $a_{i,j}$  thành  $a_{(i \bmod N) + 1, j}$ ;
- $I$  : thay thế mỗi hàng trong tất cả các hàng thành hoán vị nghịch đảo tương ứng.
- $C$  : thay thế mỗi cột trong tất cả các cột thành hoán vị nghịch đảo tương ứng.

Hoán vị nghịch đảo của hoán vị  $[p_1, p_2, \dots, p_N]$  là  $[q_1, q_2, \dots, q_N]$  mà  $p_{q_i} = i (\forall i \in [1, N])$ ;

Alice cũng chứng minh được là: *với một bảng hoán vị bất kì, khi thực hiện một trong các thao tác trên thì bảng số đó vẫn là bảng hoán vị*.

**Yêu cầu:** Cho một bảng hoán vị và một chuỗi gồm  $M$  kí tự tương ứng với các thao tác chơi liên tiếp nhau, hãy giúp Alice đưa ra giá trị của bảng sau khi thực hiện toàn bộ  $M$  thao tác đó.

## Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương  $N$  và  $M$  ( $N \leq 1000; M \leq 10^5$ ).
- Dòng thứ  $i$  trong số  $N$  dòng tiếp theo chứa  $N$  số nguyên dương  $a_{i,1}, a_{i,2}, \dots, a_{i,N}$  ( $a_{i,j} \leq N$ ).
- Dòng thứ  $N + 1$  gồm  $M$  kí tự trong số các kí tự  $\{R, L, D, U, I, C\}$  biểu diễn  $M$  thao tác liên tiếp của trò chơi.

## Kết quả

Ghi ra  $N$  dòng, mỗi dòng gồm  $N$  số nguyên dương biểu diễn bảng số sau khi thực hiện  $M$  thao tác của Alice

Các số trên một dòng cách nhau bởi dấu cách.

## Ví dụ

PERTAB.INP	PERTAB.OUT
3 2 1 2 3 2 3 1 3 1 2 DR	2 3 1 3 1 2 1 2 3
3 16 1 2 3 2 3 1 3 1 2 LDICRUCILDICRUCI	2 3 1 3 1 2 1 2 3

## Giải thích

Trong ví dụ thứ nhất:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix} \rightarrow (D) \begin{pmatrix} 3 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \rightarrow (R) \begin{pmatrix} 2 & 3 & 1 \\ 3 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

## Hạn chế

- Subtask 1 (20%):  $M \leq 10$ .
- Subtask 2 (15%): Chỉ gồm các thao tác loại  $L$  và  $R$ .
- Subtask 3 (15%): Chỉ gồm các thao tác loại  $U$  và  $D$ .
- Subtask 4 (15%): Chỉ gồm các thao tác loại  $L, R, U$  và  $D$ .
- Subtask 5 (35%): không có ràng buộc gì thêm.

## Bài 2. COMMUTER

File dữ liệu vào:	COMMUTER.INP
File kết quả:	COMMUTER.OUT
Hạn chế thời gian:	1.25 seconds
Hạn chế bộ nhớ:	256 megabytes

Charlie hiện đang sinh sống tại một thành phố với  $N$  ga tàu. Các ga được đánh số từ 1 đến  $N$ . Có  $M$  tuyến đường sắt được đánh số từ 1 đến  $M$ . Tuyến đường sắt thứ  $i$  ( $1 \leq i \leq M$ ) kết nối ga  $A_i$  và ga  $B_i$  cả hai chiều, và giá vé là  $C_i$  đồng. Nhà của Charlie ở gần ga  $S$  và hàng ngày cậu đi đến trường trung học gần ga  $T$ .

Vì chi phí đi lại thường xuyên cũng khá tốn kém nên Charlie đang lên kế hoạch mua một thẻ di chuyển hàng ngày để tiết kiệm chi phí di chuyển giữa hai ga này. Khi anh ta mua thẻ di chuyển hàng ngày, anh ta cần chọn một lộ trình từ ga  $S$  đến ga  $T$  sao cho đó là lộ trình có chi phí nhỏ nhất. Bằng việc sử dụng thẻ di chuyển hàng ngày này, anh ta có thể đi trên bất kỳ tuyến đường sắt nào nằm trong lộ trình được chọn ở bất kỳ hướng nào mà không phải trả thêm chi phí.

Vào những ngày cuối tuần, Charlie không cần đi học, thay vào đó anh ta thường xuyên đi đến các cửa hàng sách gần ga  $U$  và ga  $V$  và tất nhiên việc mua thẻ di chuyển cũng được áp dụng vào cuối tuần. Do đó, khi mua một thẻ di chuyển hàng ngày, Charlie mong muốn có thể giảm thiểu chi phí từ ga  $U$  đến ga  $V$ . Khi anh ta di chuyển từ ga  $U$  đến ga  $V$ , Charlie trước tiên chọn một tuyến đường từ ga  $S$  đến ga  $T$ . Sau đó, số tiền anh ta phải trả là:

- 0 đồng nếu tuyến đường thứ  $i$  nằm trong lộ trình được chọn khi anh ta mua thẻ di chuyển hàng ngày, hoặc
- $C_i$  đồng nếu tuyến đường thứ  $i$  không nằm trong lộ trình được chọn khi anh ta mua thẻ di chuyển hàng ngày.

Tổng số tiền trên là chi phí từ ga  $U$  đến ga  $V$ . Charlie muốn biết chi phí tối thiểu từ ga  $U$  đến ga  $V$  nếu anh ta chọn một lộ trình phù hợp khi mua thẻ di chuyển hàng ngày.

### Dữ liệu vào

- Dòng đầu tiên chứa 2 số nguyên dương  $N, M$  ( $2 \leq N \leq 100000, 1 \leq M \leq 200000$ ).
- Dòng thứ hai chứa 2 số nguyên dương  $S, T$  ( $1 \leq S \neq T \leq N$ ).
- Dòng thứ ba chứa 2 số nguyên dương  $U, V$  ( $1 \leq U \neq V \leq N, U \neq S$  hoặc  $V \neq T$ ).
- $M$  dòng tiếp theo, dòng thứ  $i$  gồm 3 số nguyên dương  $A_i, B_i, C_i$  ( $1 \leq A_i < B_i \leq N, 1 \leq C_i \leq 10^9$ ). Với mỗi cặp  $(i, j)$  ta có  $A_i \neq A_j$  hoặc  $B_i \neq B_j$ .

### Kết quả

- In ra chi phí tối thiểu từ ga  $U$  đến ga  $V$  nếu Charlie chọn một lộ trình phù hợp khi mua thẻ di chuyển hàng ngày.

### Hạn chế

- Subtask 1 (25%):  $S = U$ .
- Subtask 2 (25%): Chỉ có duy nhất 1 lộ trình với chi phí nhỏ nhất đi từ ga  $S$  đến ga  $T$ .

- Subtask 3 (25%):  $N \leq 300$ .
- Subtask 4 (25%): Không có giới hạn gì thêm.

### Ví dụ

COMMUTER.INP	COMMUTER.OUT
6 6 1 6 1 4 1 2 1 2 3 1 3 5 1 2 4 3 4 5 2 5 6 1	2
8 8 5 7 6 8 1 2 2 2 3 3 3 4 4 1 4 1 1 5 5 2 6 6 3 7 7 4 8 8	15

### Lưu ý

Ở test thứ nhất:

- Lộ trình mà Charlie khi mua thẻ là:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6$ .
- Charlie di chuyển từ ga 1 đến ga 4 theo lộ trình:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 4$  với chi phí là 2 đồng.

## Bài 3. Lầu cô đơn

File dữ liệu vào: LONELY.inp  
File kết quả: LONELY.out  
Hạn chế thời gian: 4 seconds  
Hạn chế bộ nhớ: 1024 megabytes

Trời thu đã đổ, những người cô đơn không có đôi, có cặp là những người bất hạnh nhất lúc này. Mặc dù không thể lấp đầy trái tim nhưng họ có thể tạo niềm vui khi lấp đầy cái bụng bằng việc đi ăn lẩu và tất nhiên là món lẩu cô đơn. Tại một nhà hàng chuyên lẩu cô đơn dành cho người hướng nội, có tất cả  $M$  bàn ăn được đánh số từ 1 đến  $M$  nằm trên một hàng. Theo như chính sách nhà hàng, nhân viên phải ghi chép lại vị trí bàn ăn của từng vị khách. Hiện tại đã có  $N$  vị khách vào nhà hàng và họ lần lượt ngồi ở các bàn ăn được đánh số  $a_1, a_2, \dots, a_N$ .

Tiếp theo,  $M - N$  vị khách lần lượt tiến vào nhà hàng, từng người sẽ thực hiện chọn bàn để ngồi như sau:

- Chọn ra một chuỗi dài nhất gồm liên tiếp các bàn ăn đang trống. Nếu có nhiều chuỗi dài nhất thì chọn chuỗi bên trái nhất.
- Vị khách sẽ chọn bàn ăn ở giữa chuỗi được chọn. Nếu chuỗi gồm số lượng chẵn bàn ăn trống thì vị khách sẽ chọn bàn ăn bên trái trong hai bàn ăn ở giữa.

Vì phải phục vụ số lượng lớn khách hàng nên nhân viên ghi chép đã dừng ghi từ sau vị khách thứ  $N$ . Để báo cáo chi tiết lại với ông chủ, nhân viên phải trả lời chính xác  $Q$  câu hỏi:

- Câu hỏi thứ  $i$  rằng ông chủ muốn biết vị khách thứ  $b_i$  khi tiến vào nhà hàng thì ngồi ở bàn thứ mấy.

Tuy đã biết được cách chọn bàn của từng vị khách nhưng vì số lượng quá nhiều, các nhân viên không thể trả lời hết  $Q$  câu hỏi. Điều này phải nhờ đến các bạn, hãy tính toán câu trả lời cho tất cả các câu hỏi để nhân viên có thể hoàn thành nhiệm vụ trong hôm nay.

### Dữ liệu vào

- Dòng đầu tiên chứa 3 số nguyên dương  $M, N, Q$  ( $1 \leq M \leq 10^{14}, 1 \leq N \leq 10^5, N \leq M, 1 \leq Q \leq 10^5$ ).
- Dòng thứ hai gồm  $N$  số nguyên dương  $a_1, a_2, \dots, a_N$  ( $1 \leq a_1 < \dots < a_N \leq M$ ).
- Dòng thứ ba gồm  $Q$  số nguyên dương  $b_1, b_2, \dots, b_Q$  ( $1 \leq b_1 < \dots < b_Q \leq M$ ).

### Kết quả

- Ứng với mỗi câu hỏi, hãy in ra câu trả lời tương ứng trên từng dòng.

### Hạn chế

- Subtask 1 (25%):  $M \leq 3 * 10^5$ .
- Subtask 2 (35%):  $M \leq 10^{14}, b_Q \leq 3 * 10^5$ .
- Subtask 3 (40%): Không có giới hạn gì thêm.

**Ví dụ**

LONELY.inp	LONELY.out
7 1 4 4 2 3 4 5	2 6 1 3
10 2 4 2 8 1 3 5 8	2 5 6 4

## Bài 4. Bầu cán bộ lớp

File dữ liệu vào:      `standard input`  
File kết quả:          `standard output`

Lớp bạn đang bầu ban cán sự. Những người có xếp hạng số phiếu bầu từ thứ 1 đến thứ  $x$  sẽ được chọn vào ban cán sự. Thứ hạng của một ứng viên  $i$  sẽ bằng (số ứng viên có nhiều phiếu hơn ứng viên  $i$ ) + 1. Bạn là một trong số  $n + 1$  ứng cử viên (bạn là ứng cử viên thứ  $n + 1$ ). Bạn biết được rằng ứng cử viên  $i$  đã giành được  $a_i$  phiếu bầu ( $1 \leq i \leq n$ ). Để chắc chắn rằng mình có thể vào ban cán sự, bạn có thể "hối lộ" kẹo cho người kiểm phiếu để giảm bớt số phiếu của các ứng cử viên khác. Với ứng cử viên  $i$  ( $1 \leq i \leq n$ ), bạn sẽ phải tốn  $b_i$  chiếc kẹo để người kiểm phiếu xóa bớt 1 phiếu của người  $i$ . Giả sử bạn muốn xóa  $t$  phiếu của người  $i$  thì bạn phải chi ra  $t \times b_i$  chiếc kẹo. Trong kho của bạn có  $k$  chiếc kẹo. Hãy xác định xem ban đầu bạn phải có tối thiểu bao nhiêu phiếu để tồn tại phương án "hối lộ" dùng **ít hơn**  $k$  chiếc kẹo, sao cho thứ hạng sau cùng của bạn không lớn hơn  $x$ ?

### Dữ liệu vào

Dòng đầu là ba số nguyên  $n, k, x$  ( $1 \leq n \leq 10^5, 1 \leq k \leq 10^{18}, 1 \leq x \leq n + 1$ ).

Dòng thứ  $i$  trong số  $n$  dòng tiếp theo gồm hai số nguyên  $a_i, b_i$  ( $0 \leq a_i \leq 10^9, 1 \leq b_i \leq 10^4$ ).

### Kết quả

In ra một số nguyên duy nhất là số phiếu ban đầu bạn phải có tối thiểu để tồn tại phương án "hối lộ" dùng **ít hơn**  $k$  chiếc kẹo, sao cho thứ hạng sau cùng của bạn không lớn hơn  $x$ .

### Ví dụ

standard input	standard output
2 1 1 4 9 0 6	4
5 15 2 10 5 0 5 10 3 0 9 4 3	6

### Hạn chế

- **Subtask 1** (30 điểm):  $1 \leq n \leq 20$ .
- **Subtask 2** (30 điểm):  $a_i \leq a_{i+1}$  và  $b_i \leq b_{i+1}$  với mọi  $1 \leq i \leq n$ .
- **Subtask 3** (20 điểm):  $a_i \leq 20$  với mọi  $1 \leq i \leq n$ .
- **Subtask 4** (20 điểm): Không có ràng buộc gì thêm.



## Bài 5. Thay đổi gốc

File dữ liệu vào:	standard input
File kết quả:	standard output
Hạn chế thời gian:	1 second
Hạn chế bộ nhớ:	256 megabytes

Cây trên bảng có  $n$  đỉnh và  $n - 1$  cạnh. Mỗi đỉnh  $i$  mang một giá trị  $v_i$ . Bên dưới là  $q$  truy vấn liên quan đến cây được thầy Tí viết lên bảng như sau:

- 1  $v$ : đổi nút gốc của cây thành nút  $v$ .
- 2  $u\ v\ x$ : Với mỗi nút trong cây con có độ lớn nhỏ nhất chứa cả nút  $u$  và  $v$ . Hãy tăng giá trị của các nút trong cây con đó thêm  $x$ .
- 3  $v$ : Tính tổng giá trị của cây con gốc  $v$ .

Cây con của đỉnh  $v$  là tập hợp các đỉnh sao cho  $v$  nằm trên đường đi ngắn nhất từ đỉnh này đến gốc của cây.

Hãy giúp Tí giải bài tập của thầy cho.

### Dữ liệu vào

- Dòng đầu tiên gồm hai số nguyên dương  $n$  và  $q$  ( $1 \leq n, q \leq 10^5$ ) lần lượt là số nút của cây và số truy vấn thầy Tí đưa.
- Dòng thứ hai gồm  $n$  số nguyên  $a_1, a_2, a_3, \dots, a_n$  ( $-10^8 \leq a_i \leq 10^8$ ) là các giá trị ban đầu của các nút.
- $n - 1$  dòng tiếp theo, mỗi dòng chứa hai số nguyên  $u$  và  $v$  ( $1 \leq u, v \leq n, u \neq v$ ) biểu diễn một cạnh của cây có nối từ đỉnh  $u$  tới đỉnh  $v$ .
- $q$  dòng tiếp theo biểu diễn các truy vấn, mỗi truy vấn thuộc một trong ba dạng như sau:
  - 1  $v$  ( $1 \leq v \leq n$ ): Thể hiện truy vấn loại 1.
  - 2  $u\ v\ x$  ( $1 \leq u, v \leq n, -10^8 \leq x \leq 10^8$ ): Thể hiện truy vấn loại 2.
  - 3  $v$  ( $1 \leq v \leq n$ ): Thể hiện truy vấn loại 3.

### Kết quả

- Với mỗi truy vấn 3, in ra một dòng là kết quả của truy vấn đó.

### Hạn chế

- Subtask 1 (15% số điểm):  $n, q \leq 5000$ .
- Subtask 2 (15% số điểm): mỗi đỉnh có tối đa hai cạnh kết nối với đỉnh đó.
- Subtask 3 (15% số điểm): không có truy vấn loại 1.
- Subtask 4 (15% số điểm): không có truy vấn loại 2.
- Subtask 5 (20% số điểm): trong các truy vấn loại 2:  $u = v$ .

- Subtask 6 (20% số điểm): không có ràng buộc gì thêm.

### Ví dụ

standard input	standard output
7 5	12
2 9 4 3 1 5 7	19
1 2	-10
2 3	
2 5	
3 4	
3 6	
5 7	
3 3	
1 6	
3 2	
2 1 4 -9	
3 5	

## Bài 6. Bẫy mê cung

File dữ liệu vào:	MAZETRAP.INP
File kết quả:	MAZETRAP.OUT
Hạn chế thời gian:	5 seconds
Hạn chế bộ nhớ:	256 megabytes

Mèo Tom có một mê cung rộng lớn với  $N$  căn phòng được đánh số từ 1 đến  $N$  và có  $N - 1$  đường hầm kết nối sao cho có thể đi từ bất kỳ phòng nào đến bất kỳ phòng khác. Vào một đêm nọ, chuột Jerry đã len lút len vào mê cung để phá phách. Mèo Tom rất không ưa chuột Jerry, vì vậy mèo Tom đặt một cái bẫy chuột trong phòng thứ  $T$ . Rõ ràng, chuột Jerry biết có bẫy nên tránh xa phòng thứ  $T$ , chính vì thế mèo Tom phải nghĩ ra một chiến lược tốt hơn để dụ chuột Jerry vào cái bẫy của mình đặt.

Chuột Jerry liên tục chạy xung quanh và không bao giờ dừng lại, trừ khi nó không có nơi để di chuyển. Chuột Jerry biết rằng nó để lại vết bẩn và dấu chân trong mỗi đường hầm mà nó sử dụng. Vì mắc chứng bệnh sạch sẽ, chuột sẽ không bao giờ đi qua lại một đường hầm đã bị bẩn. Để gài chuột Jerry vào bẫy, mèo Tom có thể làm sạch một đường hầm bẩn hoặc chặn một đường hầm bằng đá. Mèo Tom muốn làm điều này với số lần di chuyển tối thiểu, vì anh ta cảm thấy rất không thoải mái khi có chuột ở gần.

Chúng ta có thể mô tả điều này như một trò chơi cho hai người chơi. Chuột Jerry cố gắng tối đa hóa số lần mèo Tom thực hiện hành động. Mèo Tom cố gắng để thắng với số lần hành động tối thiểu. Người chơi đầu tiên là mèo Tom. Trên lượt của mình, mèo Tom có thể làm sạch một đường hầm bẩn hoặc chặn một đường hầm. Một đường hầm bị chặn không quan trọng việc sạch bẩn và cũng như không thể di chuyển trên đó được nữa. Tuy nhiên, nếu đến lượt hành động, mèo Tom cũng có thể chọn cách không làm gì cả và điều đó không tính vào số lần thực hiện hành động. Khi đến lượt của chuột Jerry, chuột sẽ chọn một đường hầm sạch không bị chặn và chạy đến phòng kế tiếp theo thông qua đường hầm đó. Nếu không có đường hầm nào dẫn từ phòng hiện tại, chuột Jerry sẽ không di chuyển.

Ban đầu, tất cả các đường hầm đều sạch, chuột Jerry ở trong phòng  $M$ , cái bẫy ở trong phòng  $T$  và lượt đầu tiên thuộc về mèo Tom. Số lượng lượt hành động tối thiểu mà mèo Tom cần thực hiện là bao nhiêu để đưa chuột Jerry vào bẫy nếu cả hai chơi một cách tối ưu.

### Dữ liệu vào

- Dòng đầu tiên chứa 3 số nguyên dương  $N, T, M$  ( $1 \leq N \leq 10^6; 1 \leq T, M \leq N$ );
- $N - 1$  dòng tiếp theo, mỗi dòng gồm 2 số nguyên  $u, v$  ( $1 \leq u, v \leq N$ ) – mô tả 1 đường hầm nối hai căn phòng  $u, v$ .

### Kết quả

- In ra số lượt hành động tối thiểu mèo Tom phải thực hiện.

### Hạn chế

- Subtask 1 (20%):  $N \leq 10$ .
- Subtask 2 (30%): Căn phòng  $M$  và  $T$  được kết nối trực tiếp bởi 1 đường hầm.
- Subtask 3 (20%):  $N \leq 1000$ .
- Subtask 4 (30%): Không có giới hạn gì thêm.

## Ví dụ

MAZETRAP.INP	MAZETRAP.OUT
10 1 4 1 2 2 3 2 4 3 9 3 5 4 7 4 6 6 8 7 10	4

## Lưu ý

Một phương án tối ưu như sau:

- Mèo Tom chặn đường hầm nối căn phòng 4 và 7.
- Chuột Jerry di chuyển đến căn phòng 6. Đường hầm nối căn phòng 4 và 6 bị bắn.
- Mèo Tom chặn đường hầm nối căn phòng 6 và 8.
- Chuột Jerry không di chuyển được.
- Mèo Tom làm sạch đường hầm nối căn phòng 4 và 6.
- Chuột Jerry di chuyển đến căn phòng 4. Đường hầm nối căn phòng 4 và 6 bị bắn.
- Mèo Tom chặn đường hầm nối căn phòng 2 và 3.
- Chuột Jerry di chuyển đến căn phòng 2. Đường hầm nối căn phòng 2 và 4 bị bắn.
- Mèo Tom không làm gì.
- Chuột Jerry chỉ còn cách di chuyển vào căn phòng 1 và dính bẫy.

## Bài 7. Biểu quyết

File dữ liệu vào:            **standard input**  
File kết quả:                **standard output**  
Hạn chế thời gian:        **2 seconds**  
Hạn chế bộ nhớ:           **1024 megabytes**

Ở quốc hội đang có một dự luật cần được thông qua. Mỗi đại biểu trong số  $n$  đại biểu từ 1 đến  $n$  có thể chọn ủng hộ hoặc không ủng hộ dự luật này. Có  $m$  nhóm đại biểu  $S_1, S_2, \dots, S_m$ , mỗi nhóm gồm một số đại biểu có quan hệ thân thiết. Mỗi đại biểu sẽ thuộc ít nhất một nhóm, đồng thời có thể thuộc nhiều nhóm khác nhau. Bạn cũng biết rằng vì mỗi nhóm đại biểu gồm những người thân thiết nên sẽ tồn tại một số thập phân  $p$  nằm trong đoạn  $[0, 1]$  sao cho với mọi  $1 \leq i \leq m$  có ít nhất  $\lceil p \times |S_i| \rceil$  người trong nhóm  $S_i$  ủng hộ dự luật.

Nhiệm vụ của bạn là tìm ra số  $p$  lớn nhất sao cho không thể đảm bảo tất cả  $n$  người cùng ủng hộ dự luật.

### Dữ liệu vào

Dòng đầu là hai số nguyên  $n, m$  ( $1 \leq n, m \leq 2 \times 10^5$ ).

Mỗi dòng trong số  $m$  dòng tiếp theo mô tả một nhóm đại biểu, bắt đầu bằng số đại biểu trong nhóm, sau đó là danh sách các đại biểu.

Dữ liệu đảm bảo mỗi đại biểu nằm trong ít nhất một nhóm, đồng thời tổng độ lớn của tất cả các nhóm không quá  $10^6$ .

### Kết quả

Một số thập phân duy nhất là số  $p$  lớn nhất sao cho không thể đảm bảo tất cả  $n$  người cùng ủng hộ dự luật. Kết quả được chấp nhận nếu chênh lệch với kết quả đúng không quá  $10^{-6}$ .

### Ví dụ

standard input	standard output
5 3 3 1 4 5 3 2 3 5 5 1 2 3 4 5	0.666666667
7 5 4 1 2 3 6 3 3 5 6 3 1 2 3 5 1 3 4 6 7 4 1 3 4 6	0.800000000

### Lưu ý

#### \* Subtasks:

- (1) 10% số test ứng với 10% số điểm thỏa mãn:  $1 \leq n \leq 10, 1 \leq m \leq 500$ .
- (2) 15% số test ứng với 15% số điểm thỏa mãn:  $1 \leq n, m \leq 500$ .
- (3) 75% số test ứng với 75% số điểm còn lại không có ràng buộc gì thêm.

## Bài 8. GARAGE

File dữ liệu vào:	GARAGE.INP
File kết quả:	GARAGE.OUT
Hạn chế thời gian:	1 second
Hạn chế bộ nhớ:	256 megabytes

Một trạm sửa chữa ô tô mới được xây dựng và đi vào hoạt động. Trạm sửa chữa hiện có  $N$  nhân viên được đánh số từ 1 đến  $N$ . Mỗi nhân viên đều có một trình độ chuyên môn và phụ trách các mảng khác nhau, đối với nhân viên thứ  $i$  thì trình độ của người này là  $T_i$ . Ngày hôm nay có tổng cộng  $M$  chiếc ô tô được đưa đến sửa theo thứ tự từ 1 đến  $M$ , chiếc xe thứ  $j$  có kích thước là  $F_j$ . Khi nhân viên thứ  $i$  thực hiện sửa chữa cho chiếc xe thứ  $j$  thì thời gian sửa chữa là  $T_i * F_j$  phút, một khi đã làm việc trên một chiếc xe thì nhân viên phải hoàn thành mà không được bỏ dở.

Quy trình làm việc được diễn ra theo một dây chuyền như sau: Nhân viên số 1 bắt đầu sửa chữa xe, sau khi làm xong việc của mình, chiếc xe đó được chuyển đến nhân viên số 2 phụ trách, rồi tiếp tục chuyển đến cho nhân viên số 3 ... và cho đến khi nhân viên  $N$  sửa xong thì chiếc xe được xem là hoàn thành sửa chữa. Các chiếc xe phải được sửa chữa theo thứ tự từ 1 đến  $M$ .

Để thu hút nhiều khách hàng, chủ quản lý đã đặt ra một chính sách "sửa chữa không ngừng", cụ thể khi một chiếc xe ô tô được nhân viên  $i$  sửa chữa xong thì ngay lập tức chuyển sang cho nhân viên  $i + 1$  mà không có sự trì trệ nào. Sự trì trệ ở đây xảy ra, nếu như khi nhân viên  $i$  khi giao chiếc xe cho nhân viên  $i + 1$  mà nhân viên  $i + 1$  lại đang làm việc dở trên chiếc xe khác. Điều này vi phạm đến chính sách của công ty.

Bạn là người được quản lý giao nhiệm vụ phân phối các thời điểm cho từng chiếc xe để đi vào dây chuyền sửa chữa. Hãy cho biết lượng thời gian tối thiểu để hoàn thành sửa chữa tất cả  $M$  chiếc xe là bao nhiêu.

### Dữ liệu vào

- Dòng đầu tiên chứa 3 số nguyên dương  $N, M (1 \leq N, M \leq 10^5)$ .
- $N$  dòng tiếp theo, dòng thứ  $i$  gồm số nguyên dương  $T_i (1 \leq T_i \leq 10^4)$ .
- $N$  dòng tiếp theo, dòng thứ  $j$  gồm số nguyên dương  $F_j (1 \leq F_j \leq 10^4)$ .

### Kết quả

- In ra một số nguyên là thời gian nhanh nhất để sửa chữa tất cả  $M$  chiếc xe.

## Ví dụ

GARAGE . INP	GARAGE . OUT
3 3 2 1 1 2 1 1	11
3 3 2 3 3 2 1 2	29
4 5 3 2 2 2 3 1 2 1 2	55

## Lưu ý

- Ở ví dụ đầu tiên, chiếc xe thứ nhất giao cho nhân viên số 1 và thời gian thực hiện là 4 phút.
  - Sau 4 phút sửa chữa, thì chuyển ngay cho nhân viên số 2. Chiếc xe thứ hai lúc này chưa bàn giao cho nhân viên số 1 vì sẽ xảy ra sự chậm trễ ở sau phút thứ 7 (ta có thể chạy tay để thấy).
  - Sau phút thứ 5 ta mới đưa chiếc xe thứ hai vào dây chuyền sửa chữa (đưa cho nhân viên số 1) và theo đó sau phút thứ 7 ta mới đưa chiếc xe cuối cùng vào dây chuyền sửa chữa. Lượng thời gian để hoàn thành sửa chữa  $M$  chiếc xe chính là thời điểm hoàn thành chiếc xe thứ  $M$  là sau 11 phút kể từ khi bắt đầu.

## Bài 9. Thủ thư

File dữ liệu vào:	LIBRARIAN.INP
File kết quả:	LIBRARIAN.OUT
Hạn chế thời gian:	1 second
Hạn chế bộ nhớ:	512 megabytes

An trong khi muốn đi làm thêm đã chọn làm thủ thư tại một thư viện trong thành phố. Tại đây có  $N$  kệ để sách, mỗi kệ có  $M$  ngăn chứa sách. Thư viện có tất cả  $K$  quyển sách được đánh số từ 1 đến  $K$  (và tất nhiên có thể xếp đủ  $K$  quyển này lên kệ). Ngăn sách thứ  $j$  của kệ sách thứ  $i$  gọi là ngăn  $(i, j)$ , được biểu diễn bởi một số nguyên  $a_{i,j}$ . Nếu  $a_{i,j} = 0$ , ngăn sách đang trống, ngược lại, nó đang chứa quyển sách mang chỉ số tương ứng.

Hiện tại sách trên các kệ đang bị thay đổi vị trí quá nhiều và cậu ta muốn sắp xếp lại các quyển sách như ban đầu. Ban đầu ngăn thứ  $j$  của kệ sách thứ  $i$  đang chứa quyển sách mang chỉ số  $b_{i,j}$ . Nếu  $b_{i,j} = 0$ , ngăn sách này trống. Cậu có thể sắp xếp bằng hai thao tác sau:

- Chuyển một quyển sách sang ngăn phía bên trái hoặc bên phải nó nếu ngăn bên trái hoặc bên phải đó đang trống.
- Cầm một quyển sách và chuyển sang một ngăn đang trống trên một kệ bất kì.

Là một người cẩn thận, An chỉ chuyển một quyển sách một lúc.

Nhưng do hôm qua, cậu đã phải lau dọn lại cả thư viện nên cột sống của cậu hôm nay cũng không còn ổn, vì vậy cậu muốn thực hiện thao tác loại 2 ít nhất có thể. Hãy giúp An tính được số lượng thao tác đó là bao nhiêu để cậu có thể mua thuốc cho phù hợp?

### Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương  $N, M$  ( $1 \leq N, M \leq 10^3$ );
- $N$  dòng tiếp theo, mỗi dòng chứa  $M$  số nguyên  $a_{i,j}$  ( $0 \leq a_{i,j} \leq K$ );
- $N$  dòng tiếp theo, mỗi dòng chứa  $M$  số nguyên  $b_{i,j}$  ( $0 \leq b_{i,j} \leq K$ ).

### Kết quả

- In ra kết quả bài toán, nếu không thể sắp xếp, in ra  $-1$ .

### Hạn chế

- Subtask 1 (20%):  $M \leq 100$ ; các quyển sách ở cùng kệ sách trong trạng thái ban đầu.
- Subtask 2 (40%): Các quyển sách ở cùng kệ sách trong trạng thái ban đầu.
- Subtask 3 (40%): Không có giới hạn gì thêm.



### Ví dụ

LIBRARIAN.INP	LIBRARIAN.OUT
2 4 1 0 2 0 3 5 4 0 2 1 0 0 3 0 4 5	2
3 3 1 2 3 4 5 6 7 8 0 4 2 3 6 5 1 0 7 8	4
2 2 1 2 3 4 2 3 4 1	-1

### Lưu ý

Ở ví dụ thứ nhất, An sẽ chuyển quyển sách 1 sang phải. Cầm quyển 2 và đặt vào ngăn sách thứ nhất của kệ sách thứ nhất. Cuối cùng, cậu cầm quyển 5 và đặt vào ngăn sách thứ tư của kệ sách thứ 2.

## Bài 10. Đỉnh xanh đỏ

File dữ liệu vào:        `standard input`  
File kết quả:            `standard output`  
Hạn chế thời gian:      1 second  
Hạn chế bộ nhớ:        1024 megabytes

Cho một cây  $n$  đỉnh với gốc là đỉnh 1, mỗi đỉnh trên cây sẽ nhận một trong hai màu xanh hoặc đỏ. **Độ xấu** của cây được tính bằng số cặp đỉnh  $(u, v)$  trong đó:

- $u \neq v$
- $u$  và  $v$  cùng được tô màu xanh
- $u$  là tổ tiên của  $v$  (nghĩa là  $u$  nằm trên đường đi từ 1 đến  $v$ )

Với mỗi giá trị  $m$  từ 0 đến  $n$ , hãy tính độ xấu nhỏ nhất của cây nếu bạn phải tô  $m$  đỉnh bằng màu xanh và  $n - m$  đỉnh còn lại bằng màu đỏ.

### Dữ liệu vào

Dòng đầu là số nguyên  $n$  ( $1 \leq n \leq 2 \times 10^5$ ).

Dòng thứ  $i$  trong số  $n - 1$  dòng tiếp theo gồm một số nguyên là đỉnh cha của đỉnh  $i + 1$  trên cây.

### Kết quả

In ra  $n + 1$  số nguyên trên một dòng, số thứ  $i$  là độ xấu nhỏ nhất của cây khi  $m = i$ .

### Ví dụ

standard input	standard output
5 5 4 5 1	0 0 0 1 4 8
4 3 1 2	0 0 1 3 6

### Hạn chế

- (Subtasks 1) 20% số test ứng với 20% số điểm thỏa mãn:  $n \leq 20$ .
- (Subtasks 2) 20% số test ứng với 20% số điểm thỏa mãn:  $n \leq 500$ .
- (Subtasks 3) 60% số test ứng với 60% số điểm còn lại không có ràng buộc gì thêm.

## Bài 11. Hợp nhất đoạn

File dữ liệu vào:	standard input
File kết quả:	standard output
Hạn chế thời gian:	2 seconds
Hạn chế bộ nhớ:	1024 megabytes

Cho  $n$  đoạn trên trục số, mỗi đoạn được biểu thị bởi cặp số nguyên dương  $[l, r]$  trong đó  $l < r$ . Các đoạn này đôi một không có điểm chung hoặc chỉ có điểm chung ở đầu mút. Bạn được cho  $q$  truy vấn, mỗi truy vấn gồm hai số nguyên  $(x, y)$  trong đó  $1 \leq x \leq y \leq n$ . Với mỗi truy vấn, bạn phải tìm số nguyên không âm  $k$  nhỏ nhất sao cho tồn tại cách **nối rộng** mỗi đoạn từ đoạn thứ  $x$  đến đoạn thứ  $y$  ra không quá  $k$  đơn vị để với mọi  $x \leq i < y$  đoạn thứ  $i$  có giao điểm với đoạn thứ  $i + 1$ . Lưu ý với mỗi đoạn bạn chỉ được nối rộng sang trái và sang phải một số nguyên đơn vị mà thôi.

Ví dụ, khi  $n = 3$  và có 3 đoạn là  $(1, 2), (4, 7), (12, 17)$ :

- Nếu truy vấn  $x = 1$  và  $y = 2$  thì  $k$  nhỏ nhất bằng 1, khi đó ta có thể nối đoạn  $(1, 2)$  sang phải 1 đơn vị để được đoạn  $(1, 3)$  và nối đoạn  $(4, 7)$  sang trái 1 đơn vị để được đoạn  $(3, 7)$ , lúc ấy hai đoạn  $(1, 3)$  và  $(3, 7)$  giao nhau.
- Nếu truy vấn  $x = 2$  và  $y = 3$  thì  $k$  nhỏ nhất bằng 3, khi đó ta có thể nối đoạn  $(4, 7)$  sang phải 3 đơn vị để được đoạn  $(4, 10)$  và nối đoạn  $(12, 17)$  sang trái 2 đơn vị để được đoạn  $(10, 17)$ , lúc ấy hai đoạn  $(4, 10)$  và  $(10, 17)$  giao nhau.
- Nếu truy vấn  $x = 1$  và  $y = 3$  thì  $k$  nhỏ nhất bằng 3, khi đó ta có thể nối đoạn  $(1, 2)$  sang phải 1 đơn vị để được đoạn  $(1, 3)$ , nối đoạn  $(4, 7)$  sang trái 1 đơn vị và sang phải 2 đơn vị để được đoạn  $(3, 9)$ , đồng thời nối đoạn  $(12, 17)$  sang trái 3 đơn vị để được đoạn  $(9, 17)$ , lúc ấy ba đoạn  $(1, 3), (3, 9), (9, 17)$  lần lượt giao nhau.

### Dữ liệu vào

Dòng đầu là hai số nguyên  $n$  và  $q$  ( $1 \leq n \leq 5000, 1 \leq q \leq 10^6$ ).

Dòng thứ  $i$  trong số  $n$  dòng tiếp theo mô tả đoạn thứ  $i$  gồm hai đầu mút  $l_i, r_i$  ( $1 \leq l_i < r_i \leq 10^9$ ). Lưu ý  $r_i \leq l_{i+1}$  với mọi  $1 \leq i < n$ .

Mỗi dòng trong số  $q$  dòng tiếp theo mô tả một truy vấn gồm hai số nguyên  $x, y$  ( $1 \leq x \leq y \leq n$ ).

### Kết quả

In ra  $q$  dòng, mỗi dòng gồm một số nguyên là số  $k$  nhỏ nhất tìm được cho truy vấn tương ứng.

## Ví dụ

standard input	standard output
10 7	7
4 5	2
17 18	7
21 23	0
25 26	9
29 31	6
45 51	6
61 65	
76 77	
79 81	
85 88	
3 6	
3 5	
6 9	
1 1	
1 10	
7 8	
1 3	

## Hạn chế

- **(Subtask 1)** 15% số test ứng với 15% số điểm thỏa mãn:  $1 \leq n, q \leq 2000$ ,  $l_{i+1} \leq r_i + 20$  với mọi  $1 \leq i < n$ .
- **(Subtask 2)** 15% số test ứng với 15% số điểm thỏa mãn:  $1 \leq n, q \leq 2000$ .
- **(Subtask 3)** 70% số test ứng với 70% số điểm còn lại không có ràng buộc gì thêm.

## Bài 12. Hoán đổi

File dữ liệu vào:	standard input
File kết quả:	standard output
Hạn chế thời gian:	1 second
Hạn chế bộ nhớ:	256 megabytes

Hôm nay Hiếu và em trai Hải chơi một trò chơi với hoán vị. Xét một dãy hoán vị  $p_1, p_2, \dots, p_n$  (hoán vị là một dãy số mà mỗi số nguyên từ 1 tới  $n$  được xuất hiện chính xác một lần). Trọng số của phần tử thứ  $i$  trong hoán vị là  $a_i$ .

Đầu tiên, Hiếu và Hải chọn một phần tử thứ  $i$  ( $1 \leq i < n$ ), sau đó chia dãy hoán vị làm hai phần. Hiếu sẽ lấy các phần tử của dãy hoán vị từ vị trí thứ 1 đến vị trí thứ  $i$  (lấy các phần tử có giá trị từ  $p_1$  đến  $p_i$ ), còn Hải sẽ lấy các phần tử của dãy hoán vị từ vị trí thứ  $i + 1$  đến vị trí thứ  $n$  (lấy các phần tử có giá trị từ  $p_{i+1}$  đến  $p_n$ ).

Sau đó, Hiếu và Hải thực hiện "trao đổi" các phần tử cho nhau. Cụ thể hơn, Hiếu có thể đưa cho Hải một, hoặc vài số (có thể là không đưa) bất kì trong dãy hoán vị của Hiếu cho Hải. Ngược lại, Hải cũng có thể đưa Hiếu một, hoặc vài số (cũng có thể không đưa) bất kì trong dãy hoán vị của mình cho Hiếu. Việc trao đổi số  $p_i$  sẽ tốn một chi phí là  $a_i$ .

Mục tiêu của trò chơi là trao đổi các số trong dãy hoán vị, sao cho giá trị của số lớn nhất trong các phần tử mà Hiếu sở hữu sau khi trao đổi nhỏ hơn giá trị của số nhỏ nhất trong các phần tử mà Hải sở hữu sau khi trao đổi. Trò chơi cũng được tính là thành công nếu một trong hai dãy trở thành dãy rỗng.

Ví dụ, nếu  $p = [3, 1, 2]$  và  $a = [7, 1, 5]$  thì họ có thể thực hiện trò chơi như sau: chia dãy  $p$  thành hai phần là  $[3, 1]$  và  $[2]$  rồi đổi chỗ phần tử có giá trị bằng 2 từ phần thứ hai sang phần thứ nhất (chi phí là 5). Nếu  $p = [3, 5, 1, 6, 2, 4]$ ,  $a = [9, 2, 9, 9, 3, 9]$  thì ta chia  $p$  thành hai phần là  $[3, 5, 1]$  và  $[6, 2, 4]$ , và đổi phần tử có giá trị bằng 5 từ phần thứ nhất sang phần thứ hai, đổi phần tử có giá trị bằng 2 từ phần thứ hai sang phần thứ nhất (chi phí là 5).

Bạn hãy giúp anh em nhà Hiếu tính chi phí nhỏ nhất cần sử dụng để sau khi thực hiện tất cả các thao tác, hai dãy mà anh em nhà Hiếu nhận được là thỏa mãn yêu cầu đề bài.

### Dữ liệu vào

- Dòng thứ nhất chứa một số nguyên dương  $n$  ( $2 \leq n \leq 2 \times 10^5$ ) là độ dài của hoán vị.
- Dòng thứ hai chứa  $n$  số nguyên  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ ). Dữ liệu đảm bảo dãy chứa các phần tử có giá trị từ 1 đến  $n$ , mỗi giá trị xuất hiện chính xác một lần.
- Dòng thứ ba chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

### Kết quả

- Đưa ra một số nguyên duy nhất là kết quả bài toán.

### Hạn chế

- Subtask 1 (30% số điểm):  $n \leq 500$ .
- Subtask 2 (30% số điểm):  $n \leq 5000$ .
- Subtask 3 (40% số điểm): không có ràng buộc gì thêm.

## Ví dụ

standard input	standard output
3 3 1 2 7 2 5	5
4 2 4 1 3 5 9 8 6	5
6 3 5 1 6 2 4 9 2 9 9 3 9	5