

Bài toán A. xuanquang1999 và Quá trình chọn lọc đề bài

Tệp vào:	Đầu vào chuẩn
Tệp ra:	Đầu ra chuẩn
Giới hạn thời gian:	1 giây
Giới hạn bộ nhớ:	256 megabytes

VNOI Cup là một kì thi lập trình thi đấu, yêu cầu thí sinh sử dụng các kiến thức về thuật toán và cấu trúc dữ liệu để giải các bài toán mà các vòng thi đưa ra. Tuy mới tổ chức được hai mùa, song VNOI Cup đã thu hút được đông đảo thí sinh tham dự. Có được điều này một phần là nhờ công chuẩn bị các bài toán có chất lượng cao, độc đáo, và tràn đầy thử thách đến từ ban ra đề của kì thi! Tuy nhiên việc chọn lọc ra các bài toán cho một vòng thi quả là không đơn giản. Ban ra đề của kì thi đã tạo ra quá nhiều bài toán hay dẫn đến trường hợp ban ra đề *Kuroni* vẫn loay hoay không biết chọn ra bài nào cho vòng thi cuối cùng!

Các thành viên trong ban ra đề đã đề xuất n bài toán với các chủ đề khác nhau. Các bài toán có tính độc đáo lần lượt là a_1, a_2, \dots, a_n .

Để giúp *Kuroni* giảm được số lượng các bài toán cần xem xét, đồng thời vẫn giữ được sự độc đáo của các bài, *xuanquang1999* nói rằng có thể thay một số bài bằng cách gộp chúng lại và cho ra một bài toán độc đáo hơn. Cụ thể, *xuanquang1999* có thể thay đổi một số bài toán như sau:

- Chọn ra một chỉ số i sao cho $1 \leq i < |a|$ và $a_i < a_{i+1}$.
- Tạo ra một bài toán mới có tính độc đáo là $x = a_i + a_{i+1}$.
- Xóa đi bài toán thứ i và $(i + 1)$ ra khỏi danh sách bài toán, và chèn bài toán mới có tính độc đáo x vào vị trí i .
- Sau khi xóa, các bài toán thứ $i + 2, i + 3, \dots$ sẽ có chỉ số giảm đi 1.

Bằng cách sử dụng phép biến đổi trên (không hoặc nhiều lần) một cách tối ưu, hãy giúp *xuanquang1999* tìm ra số lượng bài toán nhỏ nhất có thể đạt được để giúp *Kuroni* lựa chọn các bài một cách dễ dàng hơn!

Đầu vào

Dòng đầu tiên gồm số nguyên t ($1 \leq t \leq 10\,000$) – số lượng bộ dữ liệu. Mô tả của các bộ dữ liệu như sau.

Dòng đầu tiên của bộ dữ liệu gồm số nguyên n ($2 \leq n \leq 200\,000$) – số lượng các bài toán mà ban ra đề đã tạo ra.

Dòng thứ hai của bộ dữ liệu gồm n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) – tính độc đáo của các bài toán mà ban ra đề đã ra.

Dữ liệu đảm bảo tổng n trong tất cả các bộ dữ liệu không quá 200 000.

Đầu ra

Với mỗi bộ dữ liệu, in ra một số nguyên là số lượng bài toán nhỏ nhất có thể đạt được khi *xuanquang1999* sử dụng thao tác biến đổi trên (không hoặc nhiều lần).

Điểm

Số điểm nhận được nếu bạn giải thành công bài toán này là 1250 điểm.

Ví dụ

Đầu vào chuẩn	Đầu ra chuẩn
3	2
6	1
2 3 1 2 4 3	5
4	
1 1 1 2	
5	
1 1 1 1 1	

Chú thích

Ở ví dụ thứ nhất, một cách tối ưu để biến đổi tính độc đảo của các bài toán là như sau:

Chỉ số được chọn	Mảng a trước khi biến đổi	Mảng a sau khi biến đổi
Chọn $i = 3$:	[2, 3, 1, 2, 4, 3]	→ [2, 3, 3, 4, 3]
Chọn $i = 1$:	[2, 3, 3, 4, 3]	→ [5, 3, 4, 3]
Chọn $i = 2$:	[5, 3, 4, 3]	→ [5, 7, 3]
Chọn $i = 1$:	[5, 7, 3]	→ [12, 3]

Ở ví dụ thứ hai, một cách tối ưu để biến đổi tính độc đảo của các bài toán là như sau:

Chỉ số được chọn	Mảng a trước khi biến đổi	Mảng a sau khi biến đổi
Chọn $i = 3$:	[1, 1, 1, 2]	→ [1, 1, 3]
Chọn $i = 2$:	[1, 1, 3]	→ [1, 4]
Chọn $i = 1$:	[1, 4]	→ [5]

Ở ví dụ thứ ba, ta không thể thực hiện phép biến đổi nào để làm số lượng bài toán ít đi.

Bài toán B. TrungNotChung và Công tác chuẩn bị kì thi

Tệp vào:	Đầu vào chuẩn
Tệp ra:	Đầu ra chuẩn
Giới hạn thời gian:	1 giây
Giới hạn bộ nhớ:	256 megabytes

Để đem lại trải nghiệm tốt nhất cho thí sinh, không chỉ đầu tư vào chất lượng của đề, mà **VNOI Cup** còn chú ý đến trải nghiệm thi đấu chung của thí sinh. Ban tổ chức **VNOI Cup** đã làm việc miệt mài nhằm mang tới môi trường thi đấu tốt nhất cho thí sinh, giúp thí sinh có trải nghiệm thi đấu như đang ở nhà. Đặc biệt hơn, vào mùa **VNOI Cup** thứ XX , ban tổ chức còn cho phép thí sinh lựa chọn loại bàn và màn hình mà mình yêu thích để tham gia thi đấu!

Mùa **VNOI Cup** thứ XX có n thí sinh tham dự, được đánh số từ 1 đến n . Trùng hợp thay, ở thành phố Hà Long, nơi **VNOI Cup** diễn ra, có n hãng bàn và n hãng màn hình. Thí sinh thứ i mong muốn sử dụng bàn của hãng p_i , và sử dụng màn hình của hãng q_i ($1 \leq p_i, q_i \leq n$).

Sau khi thu thập hết các yêu cầu của thí sinh về bàn và màn hình, ban tổ chức quyết định rằng các bàn được mua ở cùng hãng thứ i phải có cùng một độ cao là t_i **nguyên dương**. Tuy nhiên để đảm bảo không có sự nhầm lẫn giữa các bàn, các bàn mua ở các hãng **khác nhau** phải có độ cao **khác nhau**. Nói cách khác, $t_i \neq t_j$ với $i \neq j$.

Tương tự như vậy, các màn hình mua ở cùng hãng thứ i phải có cùng độ cao là m_i **nguyên dương**, và không có hai màn hình nào được mua ở hai hãng khác nhau mà có cùng độ cao ($m_i \neq m_j$ với $i \neq j$).

Để đảm bảo tính công bằng cho kì thi, ban tổ chức muốn rằng sau khi đặt màn hình lên bàn của thí sinh, độ cao từ mặt đất lên điểm trên cùng của màn hình phải **bằng nhau** đối với tất cả các thí sinh. Như vậy với mỗi thí sinh, tổng độ cao của bàn và màn hình của thí sinh đó phải bằng một hằng số c , hay $t_{p_i} + m_{q_i} = c$.

TrungNotChung đã được phân công nhiệm vụ đi mua bàn và màn hình cho các thí sinh. Hãy giúp *TrungNotChung* tìm xem các chiều cao t_1, t_2, \dots, t_n của các loại bàn cần mua, và chiều cao m_1, m_2, \dots, m_n của các loại màn hình cần mua, sao cho tổng chiều cao của bàn và màn hình đối với từng thí sinh phải bằng nhau và bằng hằng số c là **nhỏ nhất** có thể để tiết kiệm chi phí mua và vận chuyển. Nếu vẫn có nhiều đáp án, hãy in ra đáp án bất kì.

Nếu không tại đáp án nào thỏa mãn, hãy xác định điều đó.

Đầu vào

Dòng đầu tiên gồm số nguyên n ($1 \leq n \leq 10^5$) – số lượng thí sinh thi vòng chung kết **VNOI Cup** mùa XX .

Dòng thứ hai chứa n số nguyên dương p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — hãng bàn mà các thí sinh lựa chọn.

Dòng thứ hai chứa n số nguyên dương q_1, q_2, \dots, q_n ($1 \leq q_i \leq n$) — hãng màn hình mà các thí sinh lựa chọn.

Đầu ra

Nếu như không có cách mua bàn thỏa mãn yêu cầu của ban tổ chức, hãy in ra NO.

Ngược lại,

- Ở dòng đầu tiên, in ra YES.
- Dòng thứ hai chứa n số nguyên dương t_1, t_2, \dots, t_n ($1 \leq t_i \leq 10^9$, $t_i \neq t_j$ với $i \neq j$) — trong đó t_i là chiều cao của bàn từ hãng thứ i mà ban tổ chức sẽ mua.

- Dòng thứ hai chứa n số nguyên dương m_1, m_2, \dots, m_n ($1 \leq m_i \leq 10^9$, $m_i \neq m_j$ với $i \neq j$) — trong đó m_i là chiều cao của màn hình từ hãng thứ i mà ban tổ chức sẽ mua.

Đáp án cần phải thỏa mãn điều kiện độ cao được đưa ra của ban tổ chức, và tổng độ cao của bàn và màn hình cho mỗi thí sinh phải bằng một hằng số c **nhỏ nhất có thể**. Nếu có nhiều đáp án thỏa mãn, hãy in ra đáp án bất kì.

Điểm

Số điểm nhận được nếu bạn giải thành công bài toán này là 1250 điểm.

Ví dụ

Đầu vào chuẩn	Đầu ra chuẩn
3 2 1 3 2 3 1	YES 1 2 3 1 2 3
2 1 1 2 1	NO
4 1 2 1 4 2 3 2 1	YES 1 2 4 3 1 3 2 4

Bài toán C. ngfam làm Hoa Tiêu

Tệp vào:	Đầu vào chuẩn
Tệp ra:	Đầu ra chuẩn
Giới hạn thời gian:	4 giây
Giới hạn bộ nhớ:	256 megabytes

Đây là một bài toán tương tác (interactive)

Sau khi đã mua xong bàn và màn hình, *TrungNotChung* cần mang bàn và màn hình đến địa điểm thi VNOI Cup. Thành phố Hà Long – nơi VNOI Cup diễn ra, có n nút giao thông được đánh số từ 1 đến n , và có $n - 1$ con đường. Từ một nút giao thông có thể đến được một nút giao thông bất kì sử dụng một số con đường trong thành phố. Như vậy đường đi ở thành phố Hà Long có cấu trúc *cây*.

Hiện tại *TrungNotChung* đang đứng tại nút giao thông thứ 1. Không may thay, điện thoại của *TrungNotChung* không bắt được tín hiệu mạng, do đó *TrungNotChung* không thể sử dụng tính năng định vị được. *TrungNotChung* không biết bản đồ của thành phố Hà Long ra sao, chỉ nhớ rằng VNOI Cup được tổ chức tại trường THPT chuyên Hà Long – nơi có vị trí đặc địa chính là một **trọng tâm** của thành phố.

Vì không biết đến địa điểm thi như thế nào, *TrungNotChung* liền gọi điện cho *ngfam* và nhờ *ngfam* chỉ đường. *ngfam* cũng không biết vị trí của *TrungNotChung* trông như thế nào, nên *ngfam* cũng không biết làm thế nào để chỉ đường thẳng đến trường THPT chuyên Hà Long được. Nhưng sau một hồi suy nghĩ, hai bạn đã thống nhất với nhau sẽ liên lạc với nhau theo mô tả sau.

Gọi vị trí mà *TrungNotChung* đang đứng là r , ban đầu $r = 1$. Coi r là gốc cây, *TrungNotChung* sẽ sử dụng các câu lệnh sau để hỏi *ngfam*.

Câu lệnh		Giải thích
adj	v	Tìm nút giao thông kề với nút giao thông r trên đường đi từ r đến v .
subtree	v	Tìm số lượng đỉnh trong cây con có gốc là nút giao thông v .
move	v	<i>TrungNotChung</i> di chuyển đến vị trí v với điều kiện v là đỉnh kề với đỉnh r . Sau đó gán $r \leftarrow v$.

Tất nhiên thời gian là vàng, do đó *TrungNotChung* cần di chuyển đến trường THPT chuyên Hà Long càng sớm càng tốt để lắp đặt các thiết bị. Chỉ với 15 555 câu lệnh, hãy giúp *TrungNotChung* tìm đến một **trọng tâm** của thành phố.

Thành phố có thể có nhiều trọng tâm, tuy nhiên *TrungNotChung* chỉ cần tìm đến một trọng tâm **bất kì**. Khi *TrungNotChung* đã đến một trọng tâm mà không phải là địa điểm thi, lúc đó *ngfam* cũng đã biết *TrungNotChung* ở đâu và sẽ đi đón bạn ấy.

Một nút giao thông được gọi là **trọng tâm** của thành phố có cấu trúc cây gồm n đỉnh, nếu như khi xóa nút giao thông này (và các cạnh nối đến nút giao thông này) này ra khỏi thành phố, các thành phần liên thông được tạo ra đều có số đỉnh không quá $\frac{n}{2}$.

Giao thức tương tác

Trong bài toán này, chương trình của BTC sẽ đóng vai trò của *ngfam* – thực hiện các câu lệnh của *TrungNotChung*, và chương trình của bạn sẽ đóng vai trò của *TrungNotChung* – đưa ra những câu lệnh để tìm được trọng tâm của thành phố.

Đầu tiên bạn cần đọc vào một số n ($1 \leq n \leq 10\,000$) – số lượng đỉnh trên cây.

Tiếp đó, bạn cần thực hiện tương tác với chương trình của BTC sử dụng các thao tác đã được đề cập ở đề bài như sau:

Câu lệnh	Tương tác	
adj	In ra	adj v ($1 \leq v \leq n$)
	Đọc vào	<ul style="list-style-type: none">Số nguyên t khi $v \neq r$ — đỉnh kề đỉnh r trên đường đi từ r đến v.-1 khi $v = r$. Bạn nên kết thúc chương trình ngay lập tức trong trường hợp này.
subtree	In ra	subtree v ($1 \leq v \leq n$)
	Đọc vào	Số nguyên s — số lượng đỉnh trên cây con gốc v .
move	In ra	move v ($1 \leq v \leq n$)
	Đọc vào	<ul style="list-style-type: none">1 khi v là đỉnh kề đỉnh r. Sau đó r sẽ được gán bằng v.-1 khi v không kề đỉnh r. Bạn nên kết thúc chương trình ngay lập tức trong trường hợp này.

Khi đã chắc chắn đỉnh r là một trọng tâm của thành phố, hãy in ra `found` và kết thúc chương trình. Câu lệnh này sẽ không tính vào giới hạn thao tác đã cho.

Nếu cây có nhiều trọng tâm, hãy di chuyển r đến trọng tâm bất kì.

Cây đã được **cố định** trước quá trình tương tác, và sẽ **không thay đổi** trong quá trình tương tác.

Sau khi in ra một câu lệnh, đừng quên xuống dòng và flush đầu ra chuẩn, nếu không bạn có thể nhận verdict `Time limit exceeded`. Để làm điều này, bạn có thể sử dụng:

- `fflush(stdout)` hoặc `cout.flush()` trong C++;
- `System.out.flush()` trong Java;
- `flush(output)` trong Pascal;
- `stdout.flush()` trong Python;
- xem tài liệu chuẩn đối với các ngôn ngữ khác.

Điểm

Subtask	Điểm	Giới hạn
1	500	$n \leq 100$
2	750	$n \leq 1000$
3	1000	Không có giới hạn gì thêm
Total	2250	

Ví dụ

Đầu vào chuẩn	Đầu ra chuẩn
2 1	move 2 found
3 1 3 1	subtree 2 adj 2 move 3 found

Chú thích

Trong ví dụ đầu tiên, thành phố Hà Long gồm duy nhất 2 nút giao thông được nối với nhau. Hai nút giao thông này đều là trọng tâm của thành phố. Do đó, ngoài cách di chuyển đến nút giao thông 2, có thể ngay lập tức đáp lại BTC khi đang ở nút giao thông 1.

Trong ví dụ thứ hai, thành phố gồm 3 nút giao thông và 2 đường đi nối 1–3 và 2–3.

- Với truy vấn đầu tiên, ta có thể kết luận được 2 là nút giao thông lá.
- Với truy vấn thứ hai, ta có thể kết luận được rằng nút giao thông 1 không kề nút giao thông 2.

Từ đây có thể kết luận rằng 3 là trọng tâm của thành phố.

Bài toán D. darkkcyan và Quy hoạch vị trí thí sinh

Tệp vào:	Đầu vào chuẩn
Tệp ra:	Đầu ra chuẩn
Giới hạn thời gian:	2 giây
Giới hạn bộ nhớ:	256 megabytes

Vòng chung kết VNOI Cup mùa XX sẽ được tổ chức trong một căn phòng có độ dài n . Trong phòng có n vị trí cách đều nhau được đánh số từ 1 đến n , và tại vị trí thứ i có thể đặt một chiếc bàn cho thí sinh, hoặc đặt một thiết bị kết nối mạng.

Với mỗi thí sinh, ta cần chọn ra một cặp vị trí (u, v) ($1 \leq u < v \leq n$), một vị trí để đặt bàn, và một vị trí đặt thiết bị kết nối mạng. Để đảm bảo kết nối đường truyền của thí sinh là tốt nhất có thể, máy tính trên bàn của thí sinh sẽ được kết nối có dây đến thiết bị mạng của thí sinh đó. Do đó khi sắp xếp vị trí cho các thí sinh cần đảm bảo các điều kiện sau:

- Không có hai thí sinh nào có bàn ở chung một vị trí.
- Không có hai thiết bị kết nối mạng nào ở chung một vị trí.
- Vị trí của các chiếc bàn và các thiết bị kết nối mạng phải khác nhau để đảm bảo an toàn.
- Khi sắp xếp vị trí cho thí sinh, không được phép tồn tại hai cặp vị trí (u, v) và (u', v') sao cho $u < u' < v < v'$. Bởi vì nếu như tồn tại hai cặp vị trí này, hai dây mạng nối hai cặp vị trí sẽ bị *giao nhau*, và ban kỹ thuật sẽ khó đi dây hơn.

darkkcyan được phân công sắp xếp vị trí cho các thí sinh và thiết bị mạng. Tất nhiên *darkkcyan* cần làm điều này một cách tối ưu sao cho càng nhiều thí sinh có thể ngồi trong phòng thi càng tốt có thể. Trong quá trình sắp xếp vị trí cho các thí sinh, có m sự kiện xảy ra. Tại sự kiện thứ i , một thí sinh mới bước vào phòng thi và được chỉ định cặp vị trí (u_i, v_i) . Dữ liệu đảm bảo rằng tại một thời điểm bất kỳ, các vị trí của các thí sinh đã ngồi vào phòng thi đảm bảo yêu cầu nói trên.

Sau mỗi sự kiện, hãy giúp *darkkcyan* tính xem có thể thêm **tối đa** bao nhiêu thí sinh vào phòng thi mà các vị trí vẫn đảm bảo yêu cầu của đề bài.

Đầu vào

Dòng đầu tiên gồm hai số nguyên n và m ($2 \leq n \leq 10^9$, $1 \leq m \leq \min\{\frac{n}{2}, 500\,000\}$) – số lượng các vị trí trong phòng thi, và số lượng sự kiện.

Dòng thứ i trong m dòng tiếp theo chứa hai số nguyên u_i và v_i ($1 \leq u_i < v_i \leq n$) – vị trí của thí sinh vào phòng trong sự kiện thứ i .

Dữ liệu đảm bảo rằng tại một thời điểm bất kỳ, các vị trí của các thí sinh đã ngồi vào phòng thi đảm bảo yêu cầu đề bài.

Đầu ra

In ra m dòng. Dòng thứ i hãy in ra một số nguyên duy nhất là số lượng các thí sinh **tối đa** có thể thêm vào phòng thi mà các vị trí vẫn đảm bảo yêu cầu của đề bài.

Điểm

Subtask	Điểm	Giới hạn
1	500	$m \leq 5000$
2	500	$u_i > u_{i+1}$ với mọi $1 \leq i < m$
3	1250	Không có giới hạn gì thêm
Tổng	2250	

Ví dụ

Đầu vào chuẩn	Đầu ra chuẩn
7 3 3 4 5 6 1 7	2 1 0
8 2 5 7 2 4	2 1
100 10 84 95 75 76 71 73 70 78 59 66 58 69 6 11 4 22 3 51 1 79	49 48 46 45 44 43 42 41 40 38

Bài toán E. Lành và Quá trình phân phát áo VNOI cup

Tệp vào:	Đầu vào chuẩn
Tệp ra:	Đầu ra chuẩn
Giới hạn thời gian:	2 giây
Giới hạn bộ nhớ:	256 megabytes

Vào mùa XX , VNOI Cup thu hút n thí sinh tới tham gia. Kỳ thi gồm m vòng, và để tăng khả năng nhận áo, tất cả các thí sinh đều tham gia đầy đủ m vòng thi. Ở mỗi vòng, các thí sinh đều có thứ hạng là một số nguyên từ 1 đến n , và không có hai thí sinh nào có cùng thứ hạng. Định nghĩa *bảng xếp hạng* cho một vòng là danh sách các thí sinh theo thứ tự có xếp hạng từ 1 đến n .

Sau khi kết thúc cả m vòng, ban tổ chức sẽ chọn ra số k và phát áo cho tất cả các thí sinh có thứ hạng không quá k ở mỗi vòng. Ban tổ chức muốn chọn ra số k **lớn nhất**, sao cho tổng số lượng thí sinh nhận được áo không quá giới hạn s cho trước.

Ví dụ,

- Nếu $n = 3$, $m = 3$, $s = 2$, bảng xếp hạng của 3 vòng lần lượt là
 - $[3, 1, 2]$ (thí sinh 3 có thứ hạng 1, thí sinh 1 có thứ hạng 2 và thí sinh 2 có thứ hạng 3),
 - $[2, 1, 3]$,
 - $[2, 3, 1]$.

Ban tổ chức sẽ chọn $k = 1$. Khi đó, số thí sinh được phát áo là 2, đó là thí sinh có chỉ số 3 và 2. Ban tổ chức không thể chọn $k = 2$ hay $k = 3$, vì khi đó cả 3 thí sinh sẽ đều nhận được áo.

- Nếu $n = 2$, $m = 2$, $s = 1$, bảng xếp hạng của 2 vòng lần lượt là
 - $[1, 2]$,
 - $[2, 1]$.

Ban tổ chức sẽ chọn $k = 0$. Khi đó, số áo được phát là 0. Ban tổ chức không thể chọn $k = 1$ hay $k = 2$, vì khi đó cả hai thí sinh sẽ đều nhận được áo.

Lành được phân công phát áo cho các thí sinh. Hiện tại, kỳ thi đã trải qua $m - 1$ vòng thi. Giả sử kết quả của vòng cuối cùng được diễn ra một cách hoàn toàn ngẫu nhiên, tức mỗi bảng xếp hạng trong $n!$ bảng xếp hạng khác nhau sẽ có xác suất xảy ra là $\frac{1}{n!}$. Hãy giúp *Lành* tính xác suất để mỗi thí sinh nằm trong danh sách nhận áo, để *Lành* có thể chuẩn bị các chiếc áo để đóng gói cho các bạn tốt hơn!

Đầu vào

Dòng đầu tiên chứa ba số nguyên dương n, m, s ($1 \leq n, m \leq 2000$, $1 \leq s \leq n$) — lần lượt là số thí sinh tham gia, số vòng của kỳ thi và số lượng thí sinh tối đa có thể nhận được áo.

Dòng thứ i trong $m - 1$ dòng tiếp theo gồm n số nguyên dương đôi một khác nhau $r_{i,1}, r_{i,2}, \dots, r_{i,n}$ ($1 \leq r_{i,j} \leq n$) — bảng xếp hạng của vòng thi thứ i .

Đầu ra

In ra n số nguyên p_1, p_2, \dots, p_n — với p_i là xác suất để thí sinh thứ i nhận được áo, modulo 998 244 353.

Đặt $M = 998\,244\,353$. Ta thấy đáp án có thể được biểu diễn dưới dạng một phân số tối giản $\frac{p}{q}$, sao cho p và q là các số nguyên và $q \not\equiv 0 \pmod{M}$. In ra số nguyên tương ứng với $p \cdot q^{-1} \pmod{M}$, hay nói cách khác, in ra số nguyên x thỏa $0 \leq x < M$ và $x \cdot q \equiv p \pmod{M}$

Điểm

Subtask	Điểm	Giới hạn
1	500	$n, m \leq 8$
2	1250	$n, m \leq 500$
3	1000	Không có giới hạn gì thêm
Tổng	2750	

Ví dụ

Đầu vào chuẩn	Đầu ra chuẩn
3 3 2 3 1 2 2 1 3	0 665496236 665496236
2 2 1 1 2	499122177 0
3 3 3 1 2 3 2 1 3	1 1 1
8 4 6 2 7 1 4 8 6 5 3 2 7 1 3 8 6 5 4 7 2 1 8 3 6 4 5	1 1 516947969 516947969 623902721 623902721 1 516947969
3 1 2	665496236 665496236 665496236

Chú thích

Ở test mẫu đầu tiên, xét tất cả các bảng xếp hạng có thể xảy ra ở vòng cuối cùng:

- [1, 2, 3]: ban tổ chức sẽ chọn $k = 0$, và không ai nhận được áo.
- [1, 3, 2]: ban tổ chức sẽ chọn $k = 0$, và không ai nhận được áo.
- [2, 1, 3]: ban tổ chức sẽ chọn $k = 1$, và những người nhận được áo là 2 và 3.
- [2, 3, 1]: ban tổ chức sẽ chọn $k = 1$, và những người nhận được áo là 2 và 3.
- [3, 1, 2]: ban tổ chức sẽ chọn $k = 1$, và những người nhận được áo là 2 và 3.
- [3, 2, 1]: ban tổ chức sẽ chọn $k = 1$, và những người nhận được áo là 2 và 3.

Vì vậy, xác suất nhận áo của người thứ 1 là 0, còn xác suất nhận áo của người thứ 2 và 3 đều là $\frac{4}{6} = \frac{2}{3}$. Vì $665\,496\,236 \cdot 3 \equiv 2 \pmod{998\,244\,353}$, ta sẽ in ra 665 496 236.

Ở test mẫu thứ hai, xét tất cả các bảng xếp hạng có thể xảy ra ở vòng cuối cùng:

- [1, 2]: ban tổ chức sẽ chọn $k = 1$, và người nhận được áo là 1.
- [2, 1]: ban tổ chức sẽ chọn $k = 0$, và không ai nhận được áo.

Vì vậy, xác suất nhận áo của người thứ 1 là $\frac{1}{2}$, còn xác suất nhận áo của người thứ 2 là 0. Vì $499\,122\,177 \cdot 2 \equiv 1 \pmod{998\,244\,353}$, ta sẽ in ra 499 122 177.

Ở test mẫu thứ ba, vì số lượng áo phát tối đa bằng với số lượng thí sinh, nên ai cũng sẽ chắc chắn nhận được áo.

Ở test mẫu thứ tư, xác suất cho mỗi thí sinh lần lượt là $[1, 1, \frac{27}{56}, \frac{27}{56}, \frac{3}{8}, \frac{3}{8}, 1, \frac{27}{56}]$.

Ở test mẫu thứ năm, vì chỉ có một vòng duy nhất nên tất cả các thí sinh đều có xác suất nhận áo bằng nhau và bằng $\frac{2}{3}$.

Bài toán F1. FireGhost và Lát cắt hoàn hảo 1

Tệp vào:	Đầu vào chuẩn
Tệp ra:	Đầu ra chuẩn
Giới hạn thời gian:	2 giây
Giới hạn bộ nhớ:	256 megabytes

Trong phiên bản này, bạn cần tìm lát cắt sao cho độ chênh lệch về diện tích của hai miếng bánh **càng nhỏ càng tốt**.

Thầy *Lành* đang phải đóng gói những chiếc áo để gửi cho các bạn nhận được giải cho kì thi VNOI Cup mùa *XX*, *tahp* cùng em gái của mình *FireGhost* cũng liền vào giúp. Cảm động trước hành động của hai anh em em, *Lành* liền chiêu đãi hai anh em. *Lành* dẫn *tahp* và *FireGhost* đến tiệm bánh nổi tiếng nhất thành phố Hà Long, nơi mà bếp trưởng *MofK* tạo ra những chiếc bánh với những hình thù thú vị mà lại rất ngon miệng. *Lành* cho phép hai anh em chọn chiếc bánh bất kì mà hai người muốn!

Sau một hồi chọn lựa, cuối cùng hai anh em đã chọn ra chiếc bánh có hình dạng là một đa giác lồi gồm n đỉnh. Hai anh em quyết định sẽ chia sẻ chiếc bánh với nhau bằng cách cắt chiếc bánh ra làm hai phần với duy nhất một lát cắt thẳng. Hai anh em rất đoàn kết, do đó hai phần bánh sau khi được cắt ra phải có **chu vi bằng nhau**.

FireGhost rất công bằng và bình đẳng, nên *FireGhost* muốn rằng hai miếng bánh sau khi được cắt ra phải có độ chênh lệch về diện tích **càng nhỏ càng tốt**.

Sau khi nghe xong yêu cầu của *FireGhost*, bếp trưởng *MofK* rất lúng túng không biết xử trí ra sao. Cho tọa độ n đỉnh của miếng bánh, hãy giúp bếp trưởng tìm lát cắt tương ứng với yêu cầu của *FireGhost*.

Đầu vào

Dòng đầu tiên gồm số nguyên n ($3 \leq n \leq 20\,000$) – số lượng đỉnh của miếng bánh.

Dòng thứ i trong n dòng tiếp theo chứa hai số nguyên x_i và y_i ($|x_i|, |y_i| \leq 10^6$) – tọa độ đỉnh thứ i của đa giác.

Dữ liệu đảm bảo:

- danh sách các điểm được cho theo chiều ngược chiều kim đồng hồ,
- không có ba điểm liên tiếp nào thẳng hàng,
- đa giác được cho là đa giác lồi,
- không có cạnh nào của đa giác có độ dài vượt quá 49.99% chu vi.

Đầu ra

In ra hai cặp số số thực (x_{f1}, y_{f1}) và (x_{f2}, y_{f2}) – tọa độ hai điểm trên cạnh miếng bánh thể hiện nút của đường cắt thỏa mãn yêu cầu của *FireGhost* – diện tích hai miếng bánh được cắt ra cần chênh lệch **càng nhỏ càng tốt**.

Định nghĩa:

- x và y là chu vi của hai phần bánh được cắt ra bởi lát cắt bạn in ra,
- S là diện tích ban đầu của miếng bánh,
- A là độ chênh lệch diện tích hai miếng bánh tạo bởi lát cắt mà bạn in ra,
- B là độ chênh lệch diện tích hai miếng bánh tạo bởi lát cắt tương ứng của bạn tổ chức.

Đáp án của bạn sẽ được cho là **đúng** nếu như:

- hai điểm mút của lát cắt được in ra cần có khoảng cách đến giới hạn của miếng bánh không quá 10^{-6} ,
- Sai số tuyệt đối giữa chu vi của hai miếng bánh không quá 10^{-6} , hay $|x - y| \leq 10^{-6}$,
- Sai số tương đối giữa A và B so với diện tích miếng bánh không quá 10^{-12} , hay $\frac{|A - B|}{S} \leq 10^{-12}$.

Điểm

Subtask	Điểm	Giới hạn
1	250	$n = 3$
2	750	$3 \leq n \leq 500$
3	750	Không có giới hạn gì thêm
Tổng	1750	

Ví dụ

Đầu vào chuẩn	Đầu ra chuẩn
3 3 5 0 7 0 3	3 5 0 5

Chú thích

Ở test ví dụ, vì miếng bánh là một tam giác cân, ta có thể dễ dàng chia miếng bánh thành 2 phần có cùng chu vi và diện tích.

Bài toán F2. Taph và Lát cắt hoàn hảo 2

Tệp vào:	Đầu vào chuẩn
Tệp ra:	Đầu ra chuẩn
Giới hạn thời gian:	4 giây
Giới hạn bộ nhớ:	256 megabytes

Trong phiên bản này, bạn cần tìm lát cắt sao cho độ chênh lệch về diện tích của hai miếng bánh **càng lớn càng tốt**.

Thầy *Lành* đang phải đóng gói những chiếc áo để gửi cho các bạn nhận được giải cho kì thi **VNOI** Cup mùa *XX*, *taph* cùng em gái của mình *FireGhost* cũng liền vào giúp. Cảm động trước hành động của hai anh em em, *Lành* liền chiêu đãi hai anh em. *Lành* dẫn *taph* và *FireGhost* đến tiệm bánh nổi tiếng nhất thành phố Hà Long, nơi mà bếp trưởng *MofK* tạo ra những chiếc bánh với những hình thù thú vị mà lại rất ngon miệng. *Lành* cho phép hai anh em chọn chiếc bánh bất kì mà hai người muốn!

Sau một hồi chọn lựa, cuối cùng hai anh em đã chọn ra chiếc bánh có hình dạng là một đa giác lồi gồm n đỉnh. Hai anh em quyết định sẽ chia sẻ chiếc bánh với nhau bằng cách cắt chiếc bánh ra làm hai phần với duy nhất một lát cắt thẳng. Hai anh em rất đoàn kết, do đó hai phần bánh sau khi được cắt ra phải có **chu vi bằng nhau**.

Anh trai lớn *taph* vì muốn nhường em, nên hai miếng bánh nên có độ chênh lệch diện tích **càng lớn càng tốt**, để nhường em gái phần to.

Sau khi nghe xong yêu cầu của *taph*, bếp trưởng *MofK* rất lúng túng không biết xử trí ra sao. Cho tọa độ n đỉnh của miếng bánh, hãy giúp bếp trưởng tìm lát cắt tương ứng với yêu cầu của *taph*.

Đầu vào

Dòng đầu tiên gồm số nguyên n ($3 \leq n \leq 20\,000$) – số lượng đỉnh của miếng bánh.

Dòng thứ i trong n dòng tiếp theo chứa hai số nguyên x_i và y_i ($|x_i|, |y_i| \leq 10^6$) – tọa độ đỉnh thứ i của đa giác.

Dữ liệu đảm bảo:

- danh sách các điểm được cho theo chiều ngược chiều kim đồng hồ,
- không có ba điểm liên tiếp nào thẳng hàng,
- đa giác được cho là đa giác lồi,
- không có cạnh nào của đa giác có độ dài vượt quá 49.99% chu vi.

Đầu ra

In ra hai cặp số số thực (x_{f1}, y_{f1}) và (x_{f2}, y_{f2}) – tọa độ hai điểm trên cạnh miếng bánh thể hiện nút của đường cắt thỏa mãn yêu cầu của *FireGhost* – diện tích hai miếng bánh được cắt ra cần chênh lệch **càng nhỏ càng tốt**.

Định nghĩa:

- x và y là chu vi của hai phần bánh được cắt ra bởi lát cắt bạn in ra,
- S là diện tích ban đầu của miếng bánh,
- A là độ chênh lệch diện tích hai miếng bánh tạo bởi lát cắt mà bạn in ra,
- B là độ chênh lệch diện tích hai miếng bánh tạo bởi lát cắt tương ứng của bạn tổ chức.

Đáp án của bạn sẽ được cho là **đúng** nếu như:

- hai điểm mút của lát cắt được in ra cần có khoảng cách đến giới hạn của miếng bánh không quá 10^{-6} ,
- Sai số tuyệt đối giữa chu vi của hai miếng bánh không quá 10^{-6} , hay $|x - y| \leq 10^{-6}$,
- Sai số tương đối giữa A và B so với diện tích miếng bánh không quá 10^{-12} , hay $\frac{|A - B|}{S} \leq 10^{-12}$.

Điểm

Subtask	Điểm	Giới hạn
1	250	$n = 3$
2	750	$3 \leq n \leq 500$
3	750	Không có giới hạn gì thêm
Tổng	1750	

Ví dụ

Đầu vào chuẩn	Đầu ra chuẩn
3	7 0
7 0	-5.8547905591911690 4.2821858387867535
-7 6	
-1 -3	

Bài toán G. MofK và Lắp đặt thiết bị

Tệp vào:	Đầu vào chuẩn
Tệp ra:	Đầu ra chuẩn
Giới hạn thời gian:	4 giây
Giới hạn bộ nhớ:	256 megabytes

TrungNotChung đã di chuyển thiết bị đến địa điểm thi VNOI Cup. Vừa lúc đó, *darkkcyan* đã lên được kế hoạch sắp xếp vị trí cho thí sinh và *Lành, tahap* và *FireGhost* vừa mới đóng gói các chiếc áo xong. Vậy là các bạn trẻ ngay lúc đó cùng nhau dỡ các thiết bị từ xe xuống, di chuyển chúng vào phòng thi và tiến hành lắp đặt luôn!

Trên xe đang có n thiết bị được đánh số từ 1 đến n . Thiết bị thứ i có trọng lượng là a_i . Để di chuyển và lắp đặt các thiết bị một cách khoa học, các bạn trẻ sẽ di chuyển các thiết bị cần lắp đặt trước vào phòng thi trước, rồi mới đến các thiết bị khác vào sau. Sau một hồi quan sát, các bạn trẻ nhận thấy rằng với mỗi thiết bị u ($2 \leq u \leq n$), tồn tại một thiết bị p_u ($1 \leq p_u < u$) cần lắp đặt **sau khi** thiết bị u đã được lắp đặt. Như vậy thiết bị 1 lúc nào cũng là thiết bị được lắp đặt cuối cùng.

Vì thân xe hẹp nên tại một thời điểm chỉ có thể mang một thiết bị ra khỏi xe. Nếu thiết bị thứ i được di chuyển ra khỏi xe là u , vậy thời gian để di chuyển vật này ra khỏi xe là $i \cdot a_u$ đơn vị thời gian, do làm càng lâu các bạn trẻ sẽ càng mệt và di chuyển chậm hơn.

MofK được phân công chỉ đạo lắp đặt các thiết bị vào phòng thi. Để đảm bảo tiến độ công việc, *MofK* cần sắp xếp và phân công các bạn trẻ dỡ và di chuyển các thiết bị vào phòng thi càng nhanh càng tốt. Cho danh sách khối lượng các thiết bị và danh sách mối liên hệ phụ thuộc giữa chúng, hãy giúp *MofK* tìm tổng thời gian tối thiểu để các bạn có thể dỡ hết đồ từ xe xuống.

Đầu vào

Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 150\,000$) — số lượng thiết bị cần lắp đặt trong phòng thi.

Dòng thứ hai chứa n số nguyên cách nhau bởi dấu cách a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^8$) — khối lượng của các thiết bị.

Dòng thứ ba chứa $n - 1$ số nguyên cách nhau bởi dấu cách p_2, p_3, \dots, p_n ($1 \leq p_i < i$) — mô tả sự phụ thuộc của các thiết bị.

Đầu ra

In ra một số nguyên duy nhất — tổng thời gian tối thiểu để các bạn có thể dỡ hết đồ từ xe xuống.

Điểm

Subtask	Điểm	Giới hạn
1	250	$a_{p_i} \leq a_i$ với mọi $2 \leq i \leq n$
2	1000	$n \leq 2000$
3	1000	Tồn tại k sao cho $3 \leq k \leq n$ và $p_k = 1$; còn lại, $p_i = i - 1$ với mọi $2 \leq i \leq n, i \neq k$
4	1000	Không có giới hạn gì thêm
Tổng	3250	

Ví dụ

Đầu vào chuẩn	Đầu ra chuẩn
8 1 4 15 9 11 5 5 2 1 2 3 3 2 5 5	210

Chú thích

Đối với ví dụ đầu tiên, kế hoạch dỡ đồ tối ưu của các bạn trẻ sẽ như sau:

$$4 \rightarrow 7 \rightarrow 8 \rightarrow 5 \rightarrow 3 \rightarrow 6 \rightarrow 2 \rightarrow 1$$

Tổng thời gian cho kế hoạch dỡ đồ này là:

$$\begin{aligned} & 1 \cdot a_4 + 2 \cdot a_7 + 3 \cdot a_8 + 4 \cdot a_5 + 5 \cdot a_3 + 6 \cdot a_6 + 7 \cdot a_2 + 8 \cdot a_1 \\ = & 1 \cdot 9 + 2 \cdot 5 + 3 \cdot 2 + 4 \cdot 11 + 5 \cdot 15 + 6 \cdot 5 + 7 \cdot 4 + 8 \cdot 1 \\ = & 210 \end{aligned}$$

Có thể chỉ ra rằng không có kế hoạch khác cho ra thời gian tối ưu hơn.

Bài toán H. Kuroi và Những chia sẻ về quá trình ra đề VNOI Cup

Tệp vào:	Đầu vào chuẩn
Tệp ra:	Đầu ra chuẩn
Giới hạn thời gian:	4 giây
Giới hạn bộ nhớ:	256 megabytes

Góp mặt trong buổi phát trực tiếp VNOI Cup ngày hôm nay có trưởng ban ra đề *Kuroi* và anh *RR* – cựu lập trình viên thi đấu, là người hai lần lọt vào chung kết thế giới ICPC và là một trong hai người Việt Nam lọt vào chung kết thế giới kì thi Facebook Hacker Cup. Vẫn còn đam mê với lập trình thi đấu, anh còn là chủ sở hữu của fanpage *Code cùng RR*, gồm các bài viết thảo luận về lập trình thi đấu với gần 10000 người theo dõi.

Ngoài việc theo dõi và bình luận về bài làm của các thí sinh, anh *RR* và *Kuroi* còn thảo luận và chia sẻ với nhau về quá trình ra đề. *Kuroi* có rất nhiều chia sẻ thú vị về cuộc thi. Ví dụ như một thành viên trong ban ra đề vô tình tự phát minh ra kĩ thuật *Slope trick* lúc nào không hay. Hay như chuyện trưởng ban ra đề *Kuroi* phải buồn lòng từ chối các bài rất hay cũng chỉ bởi vì chúng đã trùng ý tưởng với một số bài của một số cuộc thi nhiều năm về trước. Rồi thì để đảm bảo đề bài có được chất lượng cao và mang tính đồng nhất, toàn bộ đề bài đều được soạn bởi duy nhất một thành viên trong ban ra đề...

Sau một hồi trò chuyện, anh *RR* có thắc mắc:

– Với chất lượng đề có đầu tư để đạt đến chất lượng cao như vậy, em có thể chia sẻ cho anh và các bạn khán giả nghe quá trình chọn và ra đề như thế nào được không?

– Dạ! Vì ban ra đề có rất nhiều thành viên, nên số lượng bài ra cũng rất nhiều. Vì vậy khó khăn đối với chúng em không phải là việc tạo ra bài, mà là cần chọn bài nào. Ví dụ với bài A đầu tiên trong đề, chúng em đã đề xuất đến những m bài, được đánh số từ 1 đến m . Đặc biệt hơn số lượng bài toán được đề xuất m là một số **lẻ**. Bài thứ i có độ khó là a_i . Vì chỉ được phép chọn ra một bài duy nhất, chúng em cần phải loại bớt các bài toán đi. Để làm điều này chúng em đã thực hiện thao tác sau cho đến khi còn duy nhất một bài:

- Chọn ra chỉ số i ($1 < i < m$).
- Xét ba bài toán liên tiếp có độ khó a_{i-1} , a_i và a_{i+1} .
- Loại đi bài toán **dễ nhất** và bài toán **khó nhất** trong số ba bài toán đang xét. Nếu có nhiều hơn một bài có độ khó nhỏ nhất hoặc lớn nhất, loại đi một bài bất kì.
- Đánh số lại chỉ số của các bài toán từ 1 đến $m - 2$ theo thứ tự xuất hiện của chúng. Số lượng bài toán m giảm đi 2.

Với cách làm này, chúng em đảm bảo được các bài toán quá dễ hoặc quá khó sẽ được loại bỏ, và cuối cùng chỉ tồn tại duy nhất một bài. Vì đây là vòng chung kết, nên để đảm bảo kì thi tràn đầy thử thách cho các thí sinh, chúng em đã cố gắng thực hiện thao tác trên để tìm ra được bài toán có độ khó **lớn nhất có thể** ạ.

– Ô nghe có vẻ rất thú vị! Nhưng phức tạp quá nhỉ. Bây giờ ví dụ anh có một số bài toán với độ khó lần lượt là $a = [7, 8, 3, 1]$, vậy em sẽ làm thế nào?

– Anh ơi, số lượng bài toán m cần phải là số **lẻ** ạ!

– À ừ nhỉ! Vậy với $a = [7, 8, 3, 1, 5]$ đi, em sẽ tìm ra bài toán phù hợp sử dụng thao tác loại trừ trên như thế nào?

– Dạ! Đầu tiên em sẽ chọn $i = 4$ và xét ba bài toán có độ khó là $[3, 1, 5]$. Bài toán có độ khó 1 và 5 sẽ được loại đi, nên sau đó ta có danh sách $a = [7, 8, 3]$. Thực hiện thao tác này với 3 phần tử còn lại, ta

thu được $a = [7]$. Vậy bài toán chúng em chọn sẽ có độ khó là 7. Có thể thấy đây là độ khó **lớn nhất** có thể tìm được, bởi vì chắc chắn không có cách nào để lọc được ra bài toán có độ khó bằng 8.

– Wow thì ra là vậy!

Vô cùng hứng thú với cách chọn bài của bạn ra đề **VNOI** Cup năm nay, anh *RR* muốn tiếp tục đặt ra các câu hỏi tương tự. Anh *RR* đã lấy ra trong máy tính cá nhân của mình n bài toán với độ khó lần lượt là b_1, b_2, \dots, b_n . Thay vì hỏi độ khó của toàn bộ bài toán, anh *RR* sẽ hỏi q câu hỏi. Câu hỏi thứ i gồm hai số (l_i, r_i) ($1 \leq l_i \leq r_i \leq n, l_i \equiv r_i \pmod{2}$). Anh *RR* muốn biết bài toán mà *Kuroni* sẽ chọn trong trường hợp $a = [b_{l_i}, b_{l_i+1}, \dots, b_{r_i}]$ có độ khó là bao nhiêu.

Biết rằng anh *RR* muốn kiểm tra độ nhanh nhẹn và tính chính xác của bạn ra đề, tưởng bạn ra đề *Kuroni* sẵn sàng trả lời các câu hỏi. Tuy nhiên vì có quá nhiều câu hỏi và *Kuroni* đang ở trong buổi phát trực tiếp, nên việc sử dụng máy tính có hạn chế. Cho danh sách độ khó bài tập của anh *RR* và danh sách các câu hỏi, hãy giúp *Kuroni* tìm ra bài toán có độ khó **lớn nhất** khi sử dụng thao tác loại trừ mô tả ở trên.

Đầu vào

Dòng đầu tiên gồm hai số nguyên n và q ($1 \leq n, q \leq 100\,000$) — số lượng các bài toán và câu hỏi của anh *RR*.

Dòng tiếp theo chứa n số nguyên b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^9$) — độ khó các bài toán của anh *RR*.

Dòng thứ i trong q dòng tiếp theo chứa hai số nguyên l_i và r_i ($1 \leq l_i \leq r_i \leq n, r_i \equiv l_i \pmod{2}$) — tham số cho câu hỏi thứ i của anh *RR*.

Đầu ra

In ra q dòng. Dòng thứ i là độ khó **lớn nhất** của bài toán mà *Kuroni* sẽ chọn khi sử dụng thao tác loại trừ mô tả ở đề bài.

Điểm

Subtask	Điểm	Giới hạn
1	500	$n \leq 20, q \leq 1000$
2	1500	$n, q \leq 1000$
3	1500	Không có giới hạn gì thêm
Tổng	3500	

Ví dụ

Đầu vào chuẩn	Đầu ra chuẩn
7 5	7
0 7 8 3 1 5 9	7
1 7	7
2 6	8
1 3	8
3 7	
3 3	

Chú thích

Ở truy vấn thứ nhất, ta cần đáp án cho dãy bài toán $[0, 7, 8, 3, 1, 5, 9]$. Nhận thấy ta có thể thu được bài toán có độ khó là 7 như sau:

- Chọn chỉ số $i = 5$ và loại bỏ bài toán có độ khó 1 và 5. Dãy trở thành

$$[0, 7, 8, \underline{3}, 1, \underline{5}, 9] \rightarrow [0, 7, 8, \underline{3}, 9]$$

- Chọn chỉ số $i = 2$ và loại bỏ bài toán có độ khó 0 và 8. Dãy trở thành

$$[0, 7, 8, 3, 9] \rightarrow [7, 3, 9]$$

- Chọn chỉ số $i = 2$ và loại bỏ bài toán có độ khó 3 và 9. Dãy trở thành

$$[7, 3, 9] \rightarrow [7]$$

Vì không có dãy thao tác nào có thể thu lại được bài toán cuối cùng có độ khó lớn hơn 7, nên 7 chính là độ khó lớn nhất của bài toán mà *Kuroni* sẽ chọn.

Ở truy vấn thứ hai, ta cần đáp án cho dãy bài toán $[7, 8, 3, 1, 5]$. Các thao tác tìm bài toán phù hợp đã được *Kuroni* giải thích ở đề bài.