

TRUY VẤN

Cho dãy số nguyên $A = (a_1, a_2, \dots, a_n)$, ban đầu tất cả các phần tử của dãy A được đặt bằng 0. Xét dãy gồm m lệnh, mỗi lệnh thuộc một trong hai loại:

- ✿ $S(i, k)$: Đặt $a_i = k$
- ✿ $Q(i, j)$: Cho biết tổng các phần tử từ a_i tới a_j

Yêu cầu: Trả lời tất cả truy vấn Q

Dữ liệu: Vào từ file văn bản QUERYSUM.INP

- ✿ Dòng 1 chứa hai số nguyên dương $n, m \leq 10^5$
- ✿ m dòng tiếp theo, mỗi dòng mô tả một lệnh: Đầu dòng là một chữ cái $\in \{S, Q\}$ cho biết loại lệnh
 - ✿ Nếu ký tự đầu dòng là S : Tiếp theo là dấu cách và hai số nguyên dương i, k cách nhau bởi dấu cách ($i \leq n; k \leq 10^9$)
 - ✿ Nếu ký tự đầu dòng là Q : Tiếp theo là dấu cách và hai số nguyên dương i, j cách nhau bởi dấu cách ($i \leq j \leq n$)

Kết quả: Ghi ra file văn bản QUERYSUM.OUT, với mỗi lệnh Q ghi một số nguyên duy nhất là đáp số trên một dòng

QUERYSUM.INP	QUERYSUM.OUT
S 7	5
S 1 1	8
S 2 2	
S 3 3	
Q 2 4	
S 4 4	
S 3 1	
Q 1 5	

TRUY VẤN

Hệ thống quản lý nhân sự của công ty X cần quản lý thông tin về lương của n nhân viên đánh số từ 1 tới n . Lương khởi điểm của tất cả các nhân viên là 0 và hệ thống cần cung cấp hai lệnh:

- ⚙ Lệnh cập nhật $S(i, k)$: Đặt lương cho nhân viên i là k ($1 \leq i \leq n$; $0 \leq k \leq 10^9$)
- ⚙ Lệnh truy vấn $Q(i, j)$: Cho biết lương của nhân viên hưởng lương cao nhất trong số các nhân viên từ i tới j ($1 \leq i \leq j \leq n$)

Yêu cầu: Cho một dãy m lệnh thuộc một trong hai loại trên, hãy trả lời tất cả các lệnh truy vấn

Dữ liệu: Vào từ file văn bản QUERY.INP

- ⚙ Dòng 1 chứa hai số nguyên dương $n, m \leq 10^5$
- ⚙ m dòng tiếp theo, mỗi dòng chứa thông tin về một lệnh, đầu tiên là một ký tự $\in \{S, Q\}$
 - ⚙ Nếu ký tự đầu dòng là S , tiếp theo là hai số nguyên i, k cho biết lệnh đó là $S(i, k)$
 - ⚙ Nếu ký tự đầu dòng là Q , tiếp theo là hai số nguyên i, j cho biết lệnh đó là $Q(i, j)$

Kết quả: Ghi ra file văn bản QUERY.OUT

Tương ứng với mỗi lệnh truy vấn Q trong file dữ liệu, ghi ra trên một dòng một số nguyên là câu trả lời cho truy vấn đó.

Ví dụ:

QUERY.INP	QUERY.OUT
S 6	5
S 2 1	7
S 4 5	
Q 2 4	
S 3 6	
S 2 7	
Q 1 4	

QUẢN LÝ LƯƠNG

Một công ty có n người đánh số từ 1 tới n , người thứ i có lương là w_i . ($w_i \leq 10^{18}$). Tổng giám đốc công ty được đánh số 1, mỗi người từ 2 tới n có đúng 1 thủ trưởng trực tiếp của mình. Ta nói người i quản lý người j nếu tồn tại dãy $i = x_1, x_2, \dots, x_k = j$ sao cho người x_i là thủ trưởng trực tiếp của người x_{i+1} . Cơ cấu tổ chức đảm bảo rằng không tồn tại hai người a, b mà người a quản lý người b đồng thời người b quản lý người a .

Mỗi người được quyền tăng/giảm lương của tất cả mọi người trong quyền quản lý của mình. Bạn cần viết một chương trình quản lý lương xử lý hai tác vụ:

- ⚙ p A x: Người A tăng lương của tất cả những người trong quyền quản lý của mình thêm x đồng (x có thể âm, $-10^9 \leq x \leq 10^9$)
- ⚙ u A: Cho biết lương của người A

Dữ liệu: Vào từ file văn bản SALARY.INP

- ⚙ Dòng 1: Chứa hai số nguyên dương $n, m \leq 10^5$ trong đó m là số tác vụ
- ⚙ n dòng tiếp, dòng i chứa lương khởi điểm và số hiệu thủ trưởng của người i . Riêng với người 1, dòng tương ứng sẽ chỉ có lương khởi điểm
- ⚙ m dòng tiếp, mỗi dòng chứa một tác vụ

Kết quả: Ghi ra file văn bản SALARY.OUT

Với mỗi tác vụ loại u, in ra kết quả trên một dòng

Ví dụ

SALARY.INP	SALARY.OUT
6 7	7
5	9
4 1	7
3 2	5
7 3	
2 3	
3 5	
p 3 2	
p 2 4	
u 3	
u 6	
p 5 -2	
u 6	
u 1	

DÃY NGHỊCH THỂ

Cho n là một số nguyên dương và $x = (x_1, x_2, \dots, x_n)$ là một hoán vị của dãy số $(1, 2, \dots, n)$. Với $\forall i: 1 \leq i \leq n$, gọi t_i là số phần tử đứng trước giá trị i mà lớn hơn i trong dãy x . Khi đó dãy $t = (t_1, t_2, \dots, t_n)$ được gọi là dãy nghịch thể của dãy $x = (x_1, x_2, \dots, x_n)$.

Ví dụ: Với $n = 6$

Dãy $x = (3, 2, 1, 6, 4, 5)$ thì dãy nghịch thể của nó là $t = (2, 1, 0, 1, 1, 0)$

Dãy $x = (1, 2, 3, 4, 5, 6)$ thì dãy nghịch thể của nó là $t = (0, 0, 0, 0, 0, 0)$

Dãy $x = (6, 5, 4, 3, 2, 1)$ thì dãy nghịch thể của nó là $t = (5, 4, 3, 2, 1, 0)$

Vấn đề đặt ra là :

- ✿ Cho trước một dãy hoán vị x , hãy tìm dãy nghịch thể của x
- ✿ Cho trước một dãy nghịch thể t , hãy tìm dãy hoán vị nhận t làm dãy nghịch thể.

Dữ liệu: Vào từ file văn bản IVECTOR.INP gồm 3 dòng:

- ✿ Dòng 1: Chứa số nguyên dương $n \leq 10^5$.
- ✿ Dòng 2: Chứa dãy hoán vị x gồm n số x_1, x_2, \dots, x_n
- ✿ Dòng 3: Chứa dãy nghịch thể t : gồm n số t_1, t_2, \dots, t_n

Kết quả: Ghi ra file văn bản IVECTOR.OUT gồm 2 dòng:

- ✿ Dòng 1: Ghi lần lượt từng phần tử của dãy nghịch thể của x
- ✿ Dòng 2: Ghi lần lượt từng phần tử của dãy hoán vị của t

Các số trên một dòng của Input/Output files được/phải ghi cách nhau ít nhất một dấu cách

IVECTOR.INP	IVECTOR.OUT
6	0 0 0 0 0 0
1 2 3 4 5 6	3 2 1 6 4 5
2 1 0 1 1 0	

HOÁN VỊ JOSEPHUS

Tương truyền rằng Josephus và 40 chiến sĩ bị người La Mã bao vây trong một hang động. Họ quyết định tự vẫn chứ không chịu bị bắt. 41 chiến sĩ đứng thành vòng tròn và bắt đầu đếm theo một chiều vòng tròn, cứ người nào đếm đến 3 thì phải tự vẫn và người kế tiếp bắt đầu đếm lại từ 1. Josephus không muốn chết và đã chọn được một vị trí mà ông ta cũng với một người nữa là hai người sống sót cuối cùng theo luật này. Hai người sống sót sau đó đã đầu hàng và gia nhập quân La Mã (Josephus sau đó chỉ nói rằng đó là sự may mắn, hay “bàn tay của Chúa” mới giúp ông và người kia sống sót)...

Có rất nhiều truyền thuyết và tên gọi khác nhau về bài toán Josephus, trong toán học người ta phát biểu bài toán dưới dạng một trò chơi: Cho n người đứng quanh vòng tròn theo chiều kim đồng hồ đánh số từ 1 tới n . Họ bắt đầu đếm từ người thứ nhất theo chiều kim đồng hồ, người nào đếm đến m thì bị loại khỏi vòng và người kế tiếp bắt đầu đếm lại từ 1. Trò chơi tiếp diễn cho tới khi vòng tròn không còn lại người nào. Nếu ta xếp số hiệu của n người theo thứ tự họ bị loại khỏi vòng thì sẽ được một hoán vị (j_1, j_2, \dots, j_n) của dãy số $(1, 2, \dots, n)$ gọi là hoán vị Josephus (n, m) . Ví dụ với $n = 7, m = 3$, hoán vị Josephus sẽ là $(3, 6, 2, 7, 5, 1, 4)$.

Bài toán đặt ra là cho trước hai số n, m hãy xác định hoán vị Josephus (n, m) :

Dữ liệu: Vào từ file văn bản JOSEPHUS.INP gồm một dòng chứa hai số nguyên dương $n, m \leq 10^5$.

Kết quả: Ghi ra file văn bản JOSEPHUS.OUT trên một dòng các số j_1, j_2, \dots, j_n tương ứng với hoán vị Josephus tìm được.

Các số trên một dòng của Input/Output files được/phải ghi cách nhau ít nhất một dấu cách

Ví dụ

JOSEPHUS.INP	JOSEPHUS.OUT
7 3	3 6 2 7 5 1 4

CẬP NHẬT KHOẢNG, TRUY VẤN ĐIỂM

Cho dãy số nguyên $A = (a_1, a_2, \dots, a_n)$ ban đầu chỉ gồm các số 0. Xét m lệnh, mỗi lệnh thuộc một trong hai dạng:

- ✿ Lệnh cập nhật: I i, j, Δ : Tăng các phần tử từ a_i tới a_j , mỗi phần tử lên Δ đơn vị.
- ✿ Lệnh truy vấn: Q i : Cần cho biết giá trị phần tử a_i

Yêu cầu: Trả lời tất cả các truy vấn Q.

Dữ liệu: Vào từ file văn bản UQSEQ.INP

- ✿ Dòng 1 chứa hai số nguyên dương $n \leq 10^5; m \leq 10^5$
- ✿ m dòng tiếp theo, mỗi dòng chứa một lệnh theo đúng khuôn dạng kể trên. Trong tất cả các lệnh I, Δ có giá trị tuyệt đối không quá 10^5

Kết quả: Ghi ra file văn bản UQSEQ.OUT các giá trị trả lời cho các truy vấn Q theo đúng thứ tự, mỗi giá trị ghi trên một dòng

Ví dụ

UQSEQ.INP	UQSEQ.OUT
5 8	3
I 1 3 1	7
I 2 5 2	6
Q 4	
I 3 4 3	
I 1 1 4	
I 5 5 5	
Q 5	
Q 3	

DÃY CON TĂNG

Cho dãy số nguyên dương $A = (a_1, a_2, \dots, a_n)$, phần tử a_i có trọng số là w_i . Mỗi dãy $(a_{i_1}, a_{i_2}, \dots, a_{i_k})$ thỏa mãn:

$$\begin{cases} 1 \leq i_1 < i_2 < \dots < i_k \leq n \\ a_{i_1} < a_{i_2} < \dots < a_{i_k} \end{cases}$$

được gọi là một dãy con tăng của dãy A . Chú ý rằng dãy chỉ gồm duy nhất một phần tử của A cũng được gọi là một dãy con tăng của dãy A .

Yêu cầu: Trong số các dãy con tăng của dãy A hãy chỉ ra một dãy có tổng trọng số các phần tử là lớn nhất có thể.

Dữ liệu: Vào từ file văn bản IS.INP

- ✿ Dòng 1 chứa số nguyên dương $n \leq 10^5$
- ✿ Dòng 2 chứa n số nguyên dương a_1, a_2, \dots, a_n theo đúng thứ tự đó ($\forall i: a_i \leq 10^5$)
- ✿ Dòng 3 chứa n số nguyên dương w_1, w_2, \dots, w_n theo đúng thứ tự đó ($\forall i: w_i \leq 10^9$)

Kết quả: Ghi ra file văn bản IS.OUT

- ✿ Dòng 1 ghi số phần tử trong dãy con tăng tìm được (m)
- ✿ Dòng 2 ghi m chỉ số của các phần tử được chọn theo thứ tự tăng dần

Các số trên một dòng của Input/Output files được/phải ghi cách nhau ít nhất một dấu cách

Ví dụ

IS.INP	IS.OUT
10	6
1 2 3 6 4 5 9 6 7 8	1 2 3 5 6 7
11 22 33 66 44 55 999 66 77 88	

THẢ ĐIỀU

Trong một cuộc thi thả điều, ban giám khảo căn cứ vào độ cao của mỗi chiếc điều đạt được khi thả lên trời và xếp hạng cho chiếc điều đó theo một cách đặc biệt: Những chiếc điều không được thả cùng một lúc, mà theo trình tự từng chiếc một. Khi một chiếc điều được thả lên trời, ban giám khảo sẽ căn cứ vào độ cao của chiếc điều và xếp hạng cho chiếc điều đó bằng cách so độ cao của nó với độ cao của những chiếc điều đã thả trước đó. Ví dụ, giả sử độ cao của sáu chiếc điều theo thứ tự được thả như sau:

(78,24,68,40,39,89)

Chiếc đầu tiên xếp hạng 1 vì trước nó chưa có chiếc điều nào được thả. Chiếc thứ hai xếp hạng 2 vì $24 < 78$. Chiếc thứ ba cũng xếp hạng 2 vì $24 < 68 < 78$. Chiếc thứ tư xếp hạng 3 vì $24 < 40 < 68 < 78$, Chiếc thứ năm xếp hạng 4 vì $24 < 39 < 40 < 68 < 78$ và chiếc cuối cùng xếp hạng nhất với độ cao 89 và $24 < 39 < 40 < 68 < 78 < 89$. Như vậy trình tự dãy số xếp hạng được công bố sẽ là: 1 2 2 3 4 1. Tóm lại hạng của một chiếc điều bằng **số điều đã thả cao hơn nó cộng thêm 1**.

Yêu cầu: Có n chiếc điều lần lượt được thả lên trời, em hãy cho biết dãy số biểu diễn giá trị xếp hạng của n chiếc điều.

Dữ liệu: Cho trong tệp văn bản KITE.INP, gồm có:

- ✿ Dòng đầu một số nguyên $n \leq 10^5$ cho biết số chiếc điều tham gia dự thi.
- ✿ n dòng tiếp theo, mỗi dòng ghi một số nguyên dương $\leq 10^9$ mô tả độ cao của một chiếc điều, theo thứ tự mà nó được thả lên.

Kết quả: Ghi ra tệp văn bản KITE.OUT, gồm n dòng: dòng thứ i ghi số nguyên biểu diễn giá trị xếp hạng của chiếc điều thứ i tại thời điểm nó được thả lên.

Ví dụ:

KITE.INP	KITE.OUT
6	1
78	2
24	2
68	3
40	4
39	1
89	

HAI NHÀ MÁY ĐIỆN NGUYÊN TỬ

Hai nhà máy điện nguyên tử sẽ được xây dựng tại Byteland trong tương lai gần. Nhà máy điện thứ nhất có bán kính nguy hiểm là R_1 và nhà máy điện thứ hai có bán kính nguy hiểm là R_2 . Để tránh bị nhiễm phóng xạ, tất cả các hộ gia đình cách nhà máy thứ nhất $\leq R_1$ km hoặc cách nhà máy thứ hai $\leq R_2$ km sẽ phải chuyển chỗ ở.

Hai nhà máy sẽ được xây dựng cũng như các hộ gia đình đều nằm trên mặt phẳng với hệ tọa độ trục chuẩn Oxy, trong đó khoảng cách giữa hai điểm (x_1, y_1) và (x_2, y_2) được tính theo công thức:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Có q đề án xây dựng, mỗi đề án đều phải cho biết bán kính nguy hiểm của mỗi nhà máy là một cặp giá trị (R_1, R_2) . Nhiệm vụ của bạn là xác định số lượng gia đình phải chuyển chỗ ở ứng với mỗi đề án nếu đề án đó được thực thi.

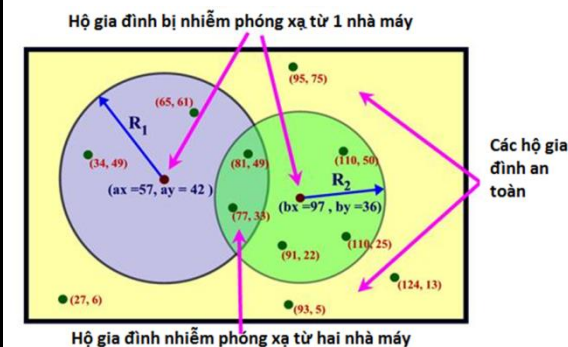
Dữ liệu: Vào từ file văn bản NUCLEAR.INP

- Dòng 1 chứa số nguyên dương n là số hộ gia đình
- n dòng tiếp, dòng i chứa hai số nguyên x_i, y_i là tọa độ (x_i, y_i) của một hộ gia đình
- Dòng $n + 2$ chứa 5 số nguyên ax, ay, bx, by, q là tọa độ của nhà máy thứ nhất (ax, ay) , tọa độ của nhà máy thứ hai (bx, by) và số đề án q .
- q dòng tiếp theo, dòng thứ j chứa hai số nguyên R_1, R_2 lần lượt là bán kính nguy hiểm của nhà máy thứ nhất và bán kính nguy hiểm của nhà máy thứ hai trong đề án thứ j .

Tất cả các số trong file dữ liệu đều là số nguyên không âm và không lớn hơn $2 \cdot 10^5$. Các số trên một dòng được ghi cách nhau ít nhất một dấu cách.

Kết quả: Ghi ra file văn bản NUCLEAR.OUT q dòng, dòng thứ j ghi một số nguyên duy nhất là số hộ gia đình phải chuyển chỗ ở nếu đề án thứ j được thực thi.

NUCLEAR.INP	NUCLEAR.OUT
11	7
95 75	7
27 6	
93 5	
124 13	
34 49	
65 61	
81 49	
77 33	
110 50	
91 22	
110 25	
57 42 97 36 2	
31 25	
25 25	



Hình minh họa với $R_1 = 31$ và $R_2 = 25$

Ít nhất 30% số điểm ứng với các test chỉ chứa các số ≤ 1000

Ít nhất 60% số điểm ứng với các test chỉ chứa các số ≤ 30000

LẬT XU

Trong giờ nghỉ, giáo sư X nghĩ ra một trò chơi “nhANH tay nhanh mắt” cho sinh viên với n đồng xu. Ban đầu n đồng xu được xếp theo thứ tự từ 1 tới n trên mặt đất, mỗi đồng xu có hai mặt: mặt ngửa của đồng xu thứ i ghi số a_i , mặt sấp của đồng xu thứ i ghi số b_i .

Phép lật với tham số C thực hiện như sau: Chọn tất cả các đồng xu có số ở mặt ngửa nhỏ hơn hoặc bằng C và lật chúng lại. Khi mỗi đồng xu bị lật, mặt ngửa trở thành mặt sấp và ngược lại, mặt sấp trở thành mặt ngửa.

Các sinh viên được cho biết trạng thái ban đầu của n đồng xu và một dãy m phép lật với các tham số c_1, c_2, \dots, c_m cho trước. Nhiệm vụ của các sinh viên là phải cho biết dãy các số ghi trên mặt ngửa của các đồng xu sau m phép lật theo đúng thứ tự đã cho.

Dữ liệu: Vào từ file văn bản FLIP.INP

- ⚙ Dòng 1 chứa hai số nguyên dương $n, m \leq 2 \cdot 10^5$
- ⚙ n dòng tiếp theo, dòng thứ i chứa hai số nguyên dương $a_i, b_i \leq 10^9$
- ⚙ m dòng tiếp theo, dòng thứ j chứa số nguyên dương $c_j \leq 10^9$

Kết quả: Ghi ra file văn bản FLIP.OUT một dòng n số nguyên là dãy các số ghi trên mặt ngửa của các đồng xu theo đúng thứ tự từ đồng xu 1 tới đồng xu n sau khi đã thực hiện m phép lật đã cho.

Các số trên một dòng của input/output files được/phải ghi cách nhau bởi dấu cách.

Ví dụ

FLIP.INP	FLIP.OUT	Giải thích
5 3	4 1 8 2 3	$\begin{array}{ccccccc} 4 & 9 & 8 & 4 & 3 \\ \hline 6 & 1 & 8 & 2 & 7 \end{array}$
4 6		Lật (8)
9 1		$\begin{array}{ccccccc} 6 & 9 & 8 & 2 & 7 \\ \hline 4 & 1 & 8 & 4 & 3 \end{array}$
8 8		Lật (2)
4 2		$\begin{array}{ccccccc} 6 & 9 & 8 & 4 & 7 \\ \hline 4 & 1 & 8 & 2 & 3 \end{array}$
3 7		Lật (9)
8		$\begin{array}{ccccccc} 4 & 1 & 8 & 2 & 3 \\ \hline 6 & 9 & 8 & 4 & 7 \end{array}$
2		
9		

