

HOÁN VỊ THỨ TỰ TỪ ĐIỂN NHỎ NHẤT

Cho dãy số nguyên $A = (a_1, a_2, \dots, a_n)$ và một số nguyên dương k , biết rằng các phần tử của dãy A là số nguyên dương trong phạm vi từ 1 tới k và mỗi giá trị từ 1 tới k đều xuất hiện trong dãy A ít nhất một lần.

Hãy tìm cách xóa đi đúng $n - k$ phần tử của dãy A và giữ nguyên thứ tự các phần tử còn lại để thu được một hoán vị có thứ tự từ điển nhỏ nhất của dãy số $(1, 2, \dots, k)$

Dữ liệu: Vào từ file văn bản SUBPERMUTE.INP

- ✿ Dòng 1 chứa số nguyên dương $T \leq 10$ là số test
- ✿ T nhóm dòng tiếp theo, mỗi nhóm dòng chứa dữ liệu một test theo khuôn dạng
 - ✿ Dòng 1 chứa hai số nguyên dương n, k ($1 \leq k \leq n \leq 10^5$)
 - ✿ Dòng 2 chứa các số nguyên a_1, a_2, \dots, a_n

Kết quả: Ghi ra file văn bản SUBPERMUTE.OUT

Ứng với mỗi test, ghi ra hoán vị còn lại sau khi xóa $n - k$ phần tử dãy A theo phương án tìm được

Ví dụ

Sample Input	Sample Output
2	3 1 2
4 3	1 3 2 4
3 2 1 2	
8 4	
4 2 3 3 1 3 2 4	

HOA VÀNG TRÊN CỎ XANH

Bộ phim “Tôi thấy HOA VÀNG trên CỎ XANH” chuyển thể từ tiểu thuyết cùng tên của nhà văn Nguyễn Nhật Ánh đang gây sốt trên màn ảnh rộng nhưng ngày thu 2015. Bờm vừa đi xem bộ phim này và khi quay về trang trại của mình, Bờm thấy một khung cảnh y hệt như trong phim.

Bản đồ trang trại là một hình chữ nhật kích thước $m \times n$ được chia làm lưới ô vuông đơn vị. Các hàng ô của lưới được đánh số từ 1 tới m theo thứ tự từ trên xuống dưới và các cột ô của lưới được đánh số từ 1 tới n theo thứ tự từ trái qua phải. Ô nằm trên hàng i và cột j được gọi là ô (i, j) , trên đó đánh dấu bởi hai con số $\in \{0,1\}$. Số 0 nghĩa là thảm cỏ còn số 1 tức là ô trồng hoa.

Bờm muốn chụp vài bức ảnh về trang trại của mình để khoe với các bạn. Máy ảnh được đặt lên một chiếc diều kéo bay qua các ô và khi diều bay qua **miền trong của ô nào**, máy ảnh sẽ tự động chụp nguyên ô đó.

Yêu cầu: Giả sử Bờm kéo diều bay từ góc trên bên trái tới góc dưới bên phải của bản đồ, cho biết máy ảnh của Bờm chụp được bao nhiêu ô trồng hoa.

Dữ liệu: Vào từ file văn bản FIELD.INP

- ✿ Dòng 1 chứa hai số nguyên dương $m, n \leq 1000$
- ✿ m dòng tiếp theo, dòng thứ i chứa n số nguyên $\in \{0,1\}$, số thứ j là con số đánh dấu ô (i, j) trên bản đồ

Các số trên một dòng của input file được ghi cách nhau bởi dấu cách.

Kết quả: Ghi ra file văn bản FIELD.OUT một số nguyên duy nhất là số ô trồng hoa được chụp

Ví dụ:

FIELD.INP	FIELD.OUT	Giải thích
6 8	9	
1 1 0 1 0 0 1 0		
1 0 0 1 1 0 0 1		
1 1 1 1 0 1 1 0		
0 0 0 1 1 1 1 1		
1 0 0 0 1 1 0 1		
0 1 0 1 0 1 1 1		

TÍCH LỚN NHẤT

Cho dãy A gồm n số nguyên a_1, a_2, \dots, a_n và một số nguyên dương $k \leq n$.

Yêu cầu: Hãy chọn ra trong dãy này đúng k phần tử sao cho tích của k phần tử này lớn nhất.

Dữ liệu: Vào từ file văn bản PRODUCT.INP:

- ✿ Dòng 1 chứa số T là số test
- ✿ T nhóm dòng tiếp theo, mỗi nhóm gồm 2 dòng ứng với một test:
 - ✿ Dòng thứ nhất chứa hai số nguyên dương n, k ($1 \leq k \leq n \leq 10^5$);
 - ✿ Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($\forall i: |a_i| \leq 10^9$)

Các số trên một dòng của input file được ghi cách nhau bởi dấu cách, số lượng các số trong một file dữ liệu không quá 2.10^5

Kết quả: Ghi ra file văn bản PRODUCT.OUT, với mỗi test ghi ra một số nguyên duy nhất trên một dòng là số dư của tích k phần tử được chọn khi chia cho 123456789

Ví dụ:

PRODUCT.INP	PRODUCT.OUT
3	60
5 3	1
1 2 3 4 5	3
6 4	
-1 -1 -1 -1 0 9	
5 3	
-1 -1 -1 2 3	

CHIA ĐOẠN

Cho một thanh gỗ độ dài L , người ta cần cắt từ thanh gỗ này n thanh gỗ với độ dài a_1, a_2, \dots, a_n , (phần thừa bỏ đi). Mỗi lần ta có thể lấy mỗi thanh gỗ cắt thành hai đoạn với tỉ lệ độ dài tùy ý, để cắt một thanh gỗ độ dài k thành hai đoạn mất chi phí đúng bằng k . Tìm cách cưa với chi phí ít nhất.

Dữ liệu: Vào từ file văn bản PARTITION.INP

Dòng 1 chứa hai số nguyên dương L, n ($L \leq 10^9; n \leq 10^5$)

Dòng 2 chứa n số nguyên dương a_1, a_2, \dots, a_n ($\sum_{i=1}^n a_i \leq L$)

Kết quả: Ghi ra file văn bản PARTITION.OUT một số nguyên duy nhất là chi phí của cách cưa tìm được

Ví dụ

PARTITION.INP	PARTITION.OUT
8 8 1 1 1 1 1 1 1 1	24

ĐƯỜNG HOA

Bản đồ giao thông của một thành phố có n địa điểm đánh số từ 1 tới n và m con đường hai chiều nối giữa các địa điểm đó. Những con đường được đánh số từ 1 tới m , con đường thứ i nối giữa hai địa điểm u_i, v_i và trên con đường này trồng loại hoa f_i . (u_i, v_i có thể trùng nhau).

Một khách du lịch muốn đi từ địa điểm s tới địa điểm t theo một hành trình thỏa mãn điều kiện: Hai con đường **liên tiếp** trên đường đi phải trồng hai loại hoa khác nhau. Hãy tìm một hành trình như vậy

Dữ liệu: Vào từ file văn bản ROADS.INP

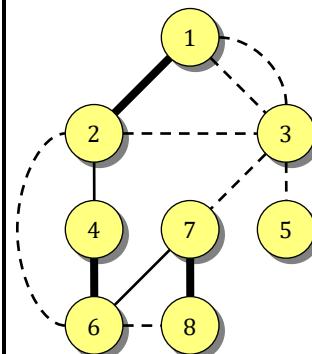
- ✿ Dòng 1 chứa bốn số nguyên dương n, m, s, t ($n, m \leq 10^5; s \neq t$)
- ✿ m dòng tiếp theo, dòng thứ i chứa ba số nguyên dương u_i, v_i, f_i ($f_i \leq 10^9$)

Kết quả: Ghi ra file văn bản ROADS.OUT

- ✿ Dòng 1 ghi số các con đường đi qua trên hành trình, nếu không tồn tại hành trình ghi số -1
- ✿ Nếu tồn tại hành trình, dòng 2 ghi số hiệu các con đường đi qua trên hành trình tìm được theo đúng thứ tự.

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

ROADS.INP	ROADS.OUT
8 12 1 8	5
1 2 1	1 12 7 11 10
1 3 3	
2 3 3	
3 1 3	
3 5 3	
3 7 3	
4 6 1	
6 2 3	
6 8 3	
7 8 1	
7 6 2	
4 2 2	
3 3 1 3	3
1 2 1	1 2 3
2 2 2	
2 3 1	



LÀM ĐỀ THI

Ngân hàng câu hỏi của trường SuperKids có 10^9 bài toán đánh số từ 1 tới 10^9 , bài toán thứ i có độ khó là i . Giáo sư X muốn chọn k bài toán khác nhau trong số đó để làm một đề thi sao cho tổng độ khó của các bài toán được chọn đúng bằng n . Hãy cho biết giáo sư X có bao nhiêu cách chọn.

(Hai cách chọn được gọi là khác nhau nếu có một bài toán được chọn trong một cách nhưng không được chọn trong cách còn lại)

Dữ liệu: Vào từ file văn bản TASKSELECT.INP

Dòng 1 ghi số nguyên dương $T \leq 10$ là số test

T dòng tiếp theo, mỗi dòng chứa hai số nguyên dương k, n ($k \leq 10; n \leq 10^9$) ứng với một test

Kết quả: Ghi ra file văn bản TASKSELECT.OUT, ứng với mỗi test, ghi ra một số nguyên duy nhất trên một dòng là số dư của kết quả tìm được khi chia cho 1000000007 ($10^9 + 7$)

Ví dụ:

TASKSELECT.INP	TASKSELECT.OUT	Giải thích
2	4	Test 1 $1 + 2 + 7 = 10$
3 10	3	$1 + 3 + 6 = 10$
2 8		$1 + 4 + 5 = 10$ $2 + 3 + 5 = 10$

BẢNG QUAN HỆ

Cho một ma trận $R = \{r_{ij}\}_{n \times n}$, các phần tử là số nguyên có giá trị tuyệt đối không quá 2. Dãy số nguyên dương $A = (a_1, a_2, \dots, a_n)$ được gọi là tương thích với ma trận R nếu:

$$\begin{cases} r_{ij} = 0 \Rightarrow a_i = a_j \\ r_{ij} = 1 \Rightarrow a_i > a_j \\ r_{ij} = -1 \Rightarrow a_i < a_j \\ r_{ij} = 2 \Rightarrow a_i \geq a_j \\ r_{ij} = -2 \Rightarrow a_i \leq a_j \end{cases}$$

Yêu cầu đặt ra là cho trước ma trận R , hãy tìm dãy số nguyên dương A tương thích với ma trận R sao cho $\max(A)$ là nhỏ nhất có thể. Biết rằng luôn tồn tại dãy tương thích với ma trận R đã cho

Dữ liệu: Vào từ file văn bản REL.INP

- ✿ Dòng 1 chứa số nguyên dương $n \leq 1000$
- ✿ n dòng tiếp theo, dòng thứ i chứa n số nguyên, số thứ j là r_{ij} .

Kết quả: Ghi ra file văn bản REL.OUT các phần tử a_1, a_2, \dots, a_n trên một dòng

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

Ví dụ

REL.INP	REL.OUT
4	2 2 1 1
0 2 1 2	
-2 0 2 2	
-1 -2 0 2	
-2 -1 -2 0	

TÌM HÌNH VUÔNG

Cho n điểm trên mặt phẳng với hệ tọa độ Đồ-các vuông góc Oxy. Các điểm được đánh số từ 1 tới n , điểm thứ i có tọa độ (x_i, y_i) . Ta nói một điểm được phủ bởi một hình vuông nếu điểm đó nằm ở miền trong của hình vuông hoặc nằm trên cạnh hình vuông.

Câu hỏi 1: Tìm số nguyên dương d nhỏ nhất sao cho có thể đặt hai hình vuông có cạnh song song với trục tọa độ, kích thước $d \times d$ để phủ hết tất cả n điểm đã cho.

Câu hỏi 2: Tìm số nguyên dương k nhỏ nhất sao cho có thể đặt hai hình vuông có cạnh song song với trục tọa độ, kích thước $k \times k$, không có điểm chung, để phủ hết tất cả n điểm đã cho.

Dữ liệu: Vào từ file văn bản SQUARE.INP

Dòng đầu tiên chứa số nguyên dương $C \leq 10$ là số bộ dữ liệu, C nhóm dòng tiếp theo, mỗi nhóm dòng mô tả một bộ dữ liệu:

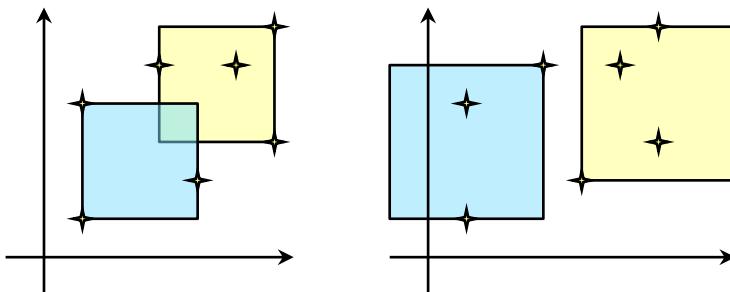
- ✿ Dòng 1 chứa số nguyên dương $n \leq 10^5$
- ✿ n dòng tiếp theo, dòng thứ i chứa hai số nguyên x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$)

Kết quả: Ghi ra file văn bản SQUARE.OUT

Ứng với mỗi bộ dữ liệu vào, ghi ra hai số nguyên dương d, k trên một dòng là đáp số cho hai câu hỏi tương ứng.

Ví dụ

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



TỔNG CÁC CHỮ SỐ

Cho số nguyên dương n , người ta viết các số nguyên liên tiếp từ 1 tới n trong hệ thập phân để tạo ra một dãy các chữ số. Hãy tính tổng các chữ số của dãy.

Ví dụ: với $n = 12$, ta có dãy số 123456789101112 với tổng là 51.

Dữ liệu: Vào từ file văn bản SUM.INP chứa duy nhất một số nguyên dương $n \leq 10^{100}$

Kết quả: Ghi ra file văn bản SUM.OUT một số nguyên duy nhất là kết quả tìm được.

Ví dụ

SUM.INP	SUM.OUT
12	51

THÁM HIỂM SA MẠC (Hard version)

Bản đồ sa mạc được vẽ trên mặt phẳng với hệ tọa độ trực chuẩn Oxy. Trên đó đánh dấu vị trí của n ốc đảo đánh số từ 1 tới n . Một nhà thám hiểm muốn đi từ ốc đảo 1 tới ốc đảo n , để tránh bị lạc đường, tại mỗi thời điểm anh ta chỉ đi tiếp theo hướng song song với một trong hai trục tọa độ

Nhà thám hiểm mang theo một bình nước dung tích là k . Khi tới mỗi ốc đảo, nhà thám hiểm sẽ đổ đầy bình nước và với bình nước đầy này anh ta có đủ nước uống để đi tiếp một quãng đường không quá k đơn vị độ dài trên bản đồ.

Yêu cầu: Xác định dung tích nhỏ nhất của bình nước cần mang theo (k) để thực hiện được hành trình từ ốc đảo 1 tới ốc đảo n và cho biết một hành trình với bình nước đó.

Dữ liệu: Vào từ file văn bản DESERT.INP

- Dòng 1: Chứa số nguyên dương n ($2 \leq n \leq 10^5$)
- n dòng tiếp theo, dòng thứ i chứa hai số nguyên x_i, y_i lần lượt là hoành độ và tung độ của ốc đảo thứ i . ($-10^9 \leq x_i, y_i \leq 10^9$)

Kết quả: Ghi ra file văn bản DESERT.OUT

- Dòng 1: Ghi số nguyên k là dung tích nhỏ nhất của bình nước cần mang theo.
- Dòng 2: Ghi số hiệu các ốc đảo đi qua theo đúng thứ tự trên hành trình bắt đầu là ốc đảo 1, kết thúc là ốc đảo n

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

Ví dụ

DESERT.INP	DESERT.OUT
8	3
1 1	1 2 3 4 6 7 8
3 2	
6 2	
7 3	
2 5	
7 5	
6 6	
4 7	

