

Chia dãy

Cho dãy số a_1, a_2, \dots, a_n . Tìm cách chia dãy a thành $s + 1$ đoạn liên tiếp sao cho tổng các số trong đoạn lớn nhất là nhỏ nhất. Đoạn lớn nhất là đoạn có tổng lớn nhất trong các đoạn được chia.

Dữ liệu: Vào từ file CHIADAY.INP

- Dòng đầu tiên chứa 2 số nguyên n, s ($n \leq 1000, s \leq 50$)
- Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^6$)

Kết quả: Ghi ra file CHIADAY.OUT một số nguyên duy nhất là tổng của đoạn lớn nhất theo cách chia tìm được.

Ví dụ:

CHIADAY.INP	CHIADAY.OUT
5 2	8
8 2 1 5 6	

Giải thích: Cách chia tối ưu là $\{8\}, \{2, 1, 5\}, \{6\}$

Ràng buộc:

- 30% số test có $s < n \leq 20, 0 \leq a_i \leq 10^6$
- 30% số test khác có $s < n \leq 1000, 0 \leq a_i \leq 10^6$
- 40% số test còn lại có $n \leq 1000, |a_i| \leq 10^6$

Three

Cho dãy số A_1, A_2, \dots, A_n ($0 \leq A_i \leq 10^6$) và số nguyên k ($k \leq n$).

Yêu cầu: Chọn ra 3 số có tích lớn nhất sao cho khoảng cách giữa vị trí của 2 số bất kỳ không vượt quá k .

Dữ liệu vào: Vào từ file **THREE.INP**

- Dòng đầu tiên chứa 2 số nguyên dương n, k ($3 \leq n \leq 10^5; 2 \leq k \leq n$)
- Dòng thứ 2 chứa n số nguyên dương a_1, a_2, \dots, a_n ($a_i \leq 10^6$)

Dữ liệu ra: ghi ra file **THREE.OUT** là tích lớn nhất của 3 số tìm được.

THREE.INP	THREE.OUT
7 3 6 1 5 2 1 3 9	60

Ràng buộc:

- 20% số test có $n \leq 10^2$
- 30% số test khác có $n \leq 5 \cdot 10^3$
- 50% số test còn lại có $n \leq 10^5$

Chuyển hộp

Một hang động nằm dưới một nút đá, nối giữa 2 thung lũng có điểm cao nhất với độ cao n có cấu trúc không bằng phẳng có thể chia thành m đoạn độ dài như nhau. Phía trên đoạn thứ i là nhũ đá độ dài a_i (tính từ điểm cao nhất của hang động). Phía dưới có những cấu trúc đá lồi lõm, ở đoạn thứ i ($1 \leq i \leq m$) có độ cao b_i . Người ta muốn thiết kế những khối hộp có mặt đáy vuông để chuyển hàng qua hang động. Vì yêu cầu đặc biệt, các khối hộp không được xoay, khi đưa vào chỉ được hoặc đẩy sang phải, hoặc di chuyển lên hoặc xuống.

Yêu cầu: Cho biết các độ cao a_1, a_2, \dots, a_m và b_1, b_2, \dots, b_m . Hãy xác định diện tích mặt bên lớn nhất của khối hộp có thể thiết kế.

Dữ liệu: Vào từ file **CAVERN.INP**

- Dòng đầu chứa 2 số nguyên n, m ($n \leq 10^9, m \leq 5000$)
- Dòng thứ 2 chứa m số nguyên không âm a_1, a_2, \dots, a_m
- Dòng thứ 3 chứa m số nguyên không âm b_1, b_2, \dots, b_m ($a_i + b_i \leq n$)

Kết quả: Ghi ra file **CAVERN.OUT** một số nguyên duy nhất là kích thước tối đa khối hộp có thể sử dụng. Trường hợp không thể có đường qua hang, đưa ra 0.

Ví dụ:

CAVERN.INP	CAVERN.OUT	Giải thích
5 6	6	...###
0 0 0 2 2 2		...###
2 0 0 0 0 0	
		#.....
		#.....

Ràng buộc:

- 30% số test có $n, m \leq 50$
- 20% số test có $n, m \leq 300$
- 20% số test khác có $m \leq 200, n \leq 10^9$
- 30% số test còn lại có $m \leq 5000, n \leq 10^9$.

Codeforce

An là 1 cao thủ lập trình thi đấu. Sau khi tu luyện nhiều năm, An có thể vào top bất cứ kỳ thi nào. Lần này, An đặt mục tiêu chinh phục tất cả các div trên codeforce. Tại đây có tất cả k div diễn ra theo chu kỳ n ngày. Ngày thứ i diễn ra contest thuộc *div* a_i . An lập một tài khoản mới codeforce và bắt đầu thi ở *div* 1. Sau mỗi lần thi, với khả năng của mình, An luôn có thể giành top và đủ điểm được thi ở *div* tiếp theo cao hơn *div* hiện tại 1 bậc. Tổng thời gian dự thi được tính bằng số contest diễn ra từ ngày An bắt đầu quyết định dự thi tới ngày An hoàn thành contest ở div k .

Yêu cầu: Hãy xác định tổng thời gian dự thi nhỏ nhất mà An có thể hoàn thiện thi xong *div* k .

Dữ liệu: Vào từ file **CODEFORCE.INP**

- Dòng đầu chứa 2 số nguyên n, k ($k \leq n \leq 2 \cdot 10^5$)
- Dòng thứ 2 chứa n số nguyên dương a, a_2, \dots, a_n ($1 \leq a_i \leq k$)

Dữ liệu đảm bảo mỗi div diễn ra ít nhất 1 contest trong chu kỳ.

Kết quả: Ghi ra file **CODEFORCE.OUT** một số nguyên duy nhất là tổng thời gian nhỏ nhất tìm được.

Ví dụ:

CODEFORCE.INP	CODEFORCE.OUT
3 3 3 2 1	5
7 5 2 1 1 4 3 2 5	19

Ràng buộc:

- 20% số test có $n \leq 10^2$
- 20% số test có $n \leq 10^3$
- 20% số test khác có $n = k$
- 40% số test còn lại có $n \leq 2 \cdot 10^5$.

Xóa cạnh

Cho đồ thị n đỉnh, m cạnh. Đỉnh thứ i có trọng số w_i . Cạnh thứ j nối giữa 2 đỉnh u_j và v_j .

Thực hiện 1 trong 2 truy vấn:

- $D\ i$: xóa cạnh thứ i của đồ thị
- $C\ u\ x$: Thay đổi trọng số đỉnh u thành x ($w_u = x$)

Yêu cầu: Sau mỗi truy vấn, hãy xác định vùng liên thông có tổng trọng số lớn nhất.

Dữ liệu: vào từ file CORRUPTED.INP

- Dòng đầu tiên chứa 3 số nguyên dương n, m, q
- Dòng thứ 2 chứa n số w_1, w_2, \dots, w_n ($w_i \leq 10^9$).
- m dòng tiếp, dòng thứ i chứa 2 số nguyên $u_i\ v_i$ ($1 \leq u_i, v_i \leq n$).
- q dòng cuối, mỗi dòng chứa thông tin 1 truy vấn loại 1 hoặc loại 2.

Kết quả: ghi ra file CORRUPTED.OUT gồm q dòng, mỗi dòng một số nguyên là tổng trọng số lớn nhất của các đỉnh trong vùng liên thông tìm được.

CORRUPTED.INP	CORRUPTED.OUT
4 4	3
1 1 1 1	5
1 2	6
2 4	6
1 4	5
2 3	
D 4	
C 3 5	
C 1 4	
D 2	
D 1	

Ràng buộc:

- 30% số test có $n, m, q \leq 1000$
- 30% số test khác, trong test chỉ có truy vấn loại C
- 40% số test còn lại có $n, m, q \leq 2 \cdot 10^5$

Hình vuông

Cho bảng vuông kích thước $m \times n$. Các hàng được đánh số $1, 2, \dots, m$ từ trên xuống dưới. Các cột được đánh số $1, 2, \dots, n$ từ trái qua phải. Tại mỗi ô của bảng có thể có một số ô cấm.

Yêu cầu: Cho Q vị trí ô trên bảng. Với mỗi ô, xác định hình vuông lớn nhất chứa ô đó và không chứa ô cấm.

Dữ liệu: Vào từ file SQUARE.INP

- Dòng đầu chứa 2 số nguyên dương m, n ($m, n \leq 2000$)
- m dòng tiếp theo, dòng thứ i chứa n ký tự biểu thị dòng i của bảng. Ký tự thứ j biểu diễn ô (i, j) . Giá trị '#' biểu diễn ô cấm.
- Dòng tiếp theo ghi số nguyên Q – số vị trí truy vấn.
- Q dòng tiếp, mỗi dòng chứa 2 số nguyên (x, y) xác định tọa độ ô truy vấn.

Kết quả: ghi ra file SQUARE.OUT gồm Q dòng, dòng thứ i ghi tương ứng diện tích hình vuông lớn nhất tìm được, chứa ô trong truy vấn thứ i .

Ví dụ:

SQUARE.INP	SQUARE.OUT
5 5	9
# . . . #	4
. . # . .	9
.	0
# . . . #	1
#	
5	
3 2	
2 5	
5 4	
4 5	
1 3	

Ràng buộc: Gọi K là số ô chứa đá.

- 15% số test tương ứng 15% số điểm có $n, m \leq 50$; $Q \leq 1000$
- 15% số test khác tương ứng 15% số điểm có $n \leq 10$; $m \leq 2000$; $Q \leq 10^5$
- 30% số test khác tương ứng 30% số điểm có $n, m \leq 800$; $K \leq 10^5$; $Q \leq 10^5$
- 40% số test còn lại tương ứng 40% số điểm có $n, m \leq 2000$; $K \leq 10^5$; $Q \leq 10^5$