

Bài A. TANK

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Xe tăng là một phương tiện có cách di chuyển rất đặc biệt. Các bánh xe của nó trải dài trên nền đất để tăng diện tích tiếp xúc, từ đó giảm áp lực lên nền. Giả sử xe tăng đang muốn đi từ A đến B, ta có thể chia đoạn đất này thành n đoạn nhỏ, đoạn thứ i có độ cứng a_i . Một xe tăng chiều dài L , khối lượng M có thể đi qua nếu tại mọi thời điểm, nó luôn đứng trên vùng đất có tổng độ cứng lớn hơn M (có nghĩa là mọi đoạn con liên tiếp độ dài L của dãy a đều phải có tổng lớn hơn hoặc bằng M). Cho biết khối lượng M của xe tăng, hãy tính chiều dài L nhỏ nhất có thể có của nó để xe tăng đi qua được vùng đất này.

Dữ liệu vào

- Dòng đầu chứa: M n
- Dòng tiếp theo chứa: a_1 a_2 ... a_n

Dữ liệu đảm bảo tổng của mảng a lớn hơn hoặc bằng M

Kết quả

Một số nguyên duy nhất là chiều dài ngắn nhất có thể của xe tăng

Ví dụ

stdin	stdout
6 5 3 2 1 4 5	3

Hạn chế

- $1 \leq n \leq 10^5$, $1 \leq a_i, M \leq 10^9$
- $n \leq 1000$ với ít nhất 50% test

Bài B. LOVE

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Ngưu Lang là vị thần chăn trâu của Ngọc Hoàng Thượng đế, vì say mê một tiên nữ phụ trách việc dệt vải tên là Chức Nữ nên bỏ bê việc chăn trâu, để trâu đi nghênh ngang vào điện Ngọc Hư. Chức Nữ cũng vì mê tiếng tiêu của Ngưu Lang nên trễ nải việc dệt vải. Ngọc Hoàng giận dữ, bắt cả hai phải ở cách xa nhau, người đầu sông Ngân, kẻ cuối sông. Mỗi năm họ chỉ được gặp nhau một lần vào rằm tháng 7 âm lịch. Đó là trong truyền thuyết, còn trong thời đại 4.0 này câu chuyện có phần hơi khác.

Theo đó, thiên đình có n khu được đánh số từ 1 đến n , Ngưu Lang ở khu 1 còn Chức Nữ ở khu n . Cầu Ô Thước không phải chỉ có 1 mà là có m , cầu thứ i nối khu x_i với khu y_i và được xây dựng xong vào ngày thứ t_i . Vậy là cầu được xây chứ không phải do chim quạ và chim khách xếp thành nữa. Do đó họ cũng không nhất thiết phải gặp nhau vào rằm tháng 7, mà sẽ gặp vào một thời điểm sớm nhất trong năm sao cho tại thời điểm đó, những cây cầu đã xây xong đảm bảo tồn tại một đường đi từ 1 đến n .

Ngưu Lang vốn lập trình giỏi, nhưng do vướng vào yêu đương nên trình độ giảm sút đi nhiều. Anh ấy muốn nhờ bạn tính giúp thời điểm sớm nhất mà họ có thể gặp nhau, biết rằng chắc chắn tồn tại thời điểm như thế.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương: n m
- Dòng thứ i trong số m dòng tiếp theo chứa: x_i y_i t_i

Kết quả

Ghi thời điểm sớm nhất họ gặp nhau

Ví dụ

stdin	stdout
5 6 1 4 6 2 5 5 3 4 2 2 1 2 2 3 3 4 5 4	4

Hạn chế

- $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$, $1 \leq t \leq 10^9$
- Có 50% số test với $n, m, t \leq 100$

Bài C. SFROG

File dữ liệu vào: NULL
File kết quả: NULL
Hạn chế thời gian: 1 giây

Nhiệm vụ của bạn trong bài này là bắn chết một con ếch. Để dễ hình dung, con ếch đang ở tọa độ x_f nào đó trên trục số nguyên. Ban đầu $-10^9 \leq x_f \leq 10^9$. Có hai thao tác có thể được thực hiện là

- Bắn vào tọa độ x
- Quan sát phạm vi $[L, H]$ để biết con ếch có ở trong phạm vi đó hay không

Mỗi lần bạn quan sát thấy con ếch, nó cũng sẽ thấy bạn và sẽ nhảy sang một vị trí khác, kề với vị trí đang đứng, tức là $x_f + 1$ hoặc $x_f - 1$. Việc nhảy sẽ diễn ra ngay sau khi bạn nhìn thấy nó

Hệ thống cung cấp thư viện "SFROG.h" để tương tác với máy, có sẵn các hàm sau:

- `int query(int L, int H)` trả về 1 hoặc 0 tương ứng thấy hoặc không thấy ếch khi quan sát đoạn $[L, H]$ ($-2 \times 10^9 \leq L \leq H \leq 2 \times 10^9$). Nếu ếch đang đứng trong đoạn $[L, H]$, kết quả sẽ trả về 1, và ếch sẽ nhảy đến vị trí kề với vị trí nó đang đứng. Ngược lại, kết quả trả ra 0 và ếch vẫn đứng yên
- `void shoot(int x)` bắn vào x ($-2 \times 10^9 \leq x \leq 2 \times 10^9$)

Khi hàm `shoot(x)` được gọi (duy nhất 1 lần), chương trình sẽ tự ngắt và chấm điểm. Bạn sẽ nhận được điểm khi bắn trúng vị trí mà ếch đang đứng. Chương trình bị chấm là "Sai kết quả" nếu:

- Không gọi hàm `shoot(x)` hoặc gọi nhưng bắn không trúng
- Gọi đến hàm `query(L, H)` quá 40 lần
- Gọi đến một trong hai hàm nhưng tham số truyền vào không thỏa mãn điều kiện được kiểm tra

Để sử dụng được thư viện, phải có dòng khai báo `#include "SFROG.h"` ở đầu chương trình.

Bài D. ARCHE

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Toán học đã xuất hiện ngay từ thuở bình minh của loài người. Gần đây các nhà khảo cổ học đã tìm thấy m viên đá có khắc các thông tin về một dãy số bí ẩn. Thông tin ghi trên mỗi viên đá đều có dạng $a_i = a_j$ hoặc $a_i \neq a_j$. Sau khi nghiên cứu kỹ, họ nhận ra rằng các thông tin này có thể mâu thuẫn với nhau. Ví dụ 3 viên đá với thông tin $a_1 = a_2$, $a_2 = a_3$, $a_3 \neq a_1$ sẽ gây mâu thuẫn. Điều này có thể do sự không nguyên vẹn qua thời gian của các mẫu vật, hoặc sai sót trong việc khai quật gây ra.

Để khắc phục, họ sẽ phải sửa một vài thông tin. Rõ ràng là dấu $=$ có thể sửa thành dấu \neq một cách dễ dàng, do đó họ sẽ chọn ra một vài viên đá có dấu $=$ và khắc thêm một nét sửa thành dấu \neq . Việc khắc đá đòi hỏi phải có dụng cụ, và viên đá càng cứng thì dụng cụ càng phải sắc bén. Họ cần phải chuẩn bị một dụng cụ sắc đến cỡ nào đây? Bạn hãy giúp các nhà khảo cổ học tìm ra các viên đá để khắc thêm, sao cho độ cứng của viên đá cứng nhất được chọn là nhỏ nhất có thể. Tất nhiên việc chọn phải đảm bảo sau khi khắc thì các thông tin không còn mâu thuẫn nữa.

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương n, m với n là số phần tử của dãy số bí ẩn
- m dòng tiếp theo mô tả m viên đá, mỗi dòng ghi ba số nguyên i, j, k ($1 \leq i, j \leq n$; $-1 \leq k \leq 10^9$). Nếu $k = -1$ thì thông tin được ghi là $a_i \neq a_j$ và ta không cần biết độ cứng của viên đá. Nếu $k \geq 0$ thì thông tin được ghi là $a_i = a_j$ và độ cứng của viên đá là k .

Kết quả

- Ghi độ cứng của viên đá cứng nhất trong cách chọn. Nếu các thông tin đã cho không mâu thuẫn thì ghi 0. Nếu không tồn tại cách chọn thỏa mãn thì ghi -1

Ví dụ

stdin	stdout
10 3 1 2 4 2 3 6 3 1 -1	4

Hạn chế

- Có 30% test với $1 \leq n, m \leq 20$
- Có 30% test với $21 \leq n, m \leq 1000$
- Có 40% test với $1001 \leq n, m \leq 10^5$