**MẠNG ĐIỆN (electronic):**

Cho một mạng điện gồm N bóng đèn được nối với nhau bởi M dây dẫn, trong đó có 2 bóng đèn được thay bằng 2 điểm đặc biệt là cực dương P và cực âm Q của mang điện.

Một bóng đèn được cho là có thể sáng được nếu tồn tại 1 đường đi đơn từ P đến Q mà đi qua bóng đèn đó. Hãy cho biết có bao nhiêu bóng đèn có thể phát sáng?

Sử dụng kiến thức về khớp, kiến thức này mình chưa hướng dẫn! :D

Inp:

- N<=10^6;

- M<=10^6;

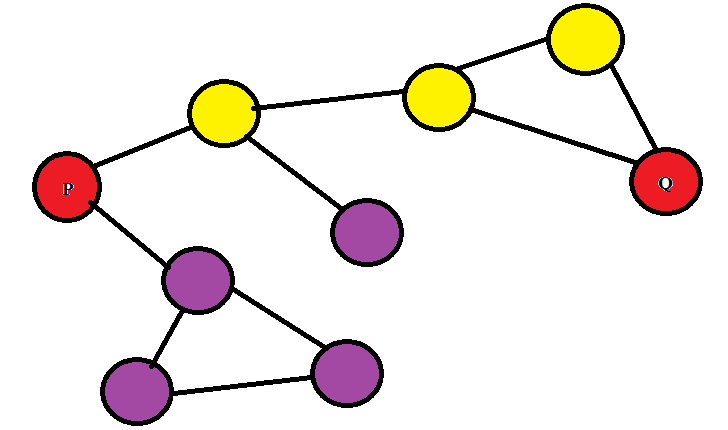
- P,Q<=N; P<>Q;

- M dòng, mỗi dòng chứa 2 số u, v chỉ ra rằng có dây nối giữa u và v

Out:

- dòng đầu là số đèn có thể phát sáng

- cho biết mã số các bóng đèn này



**TẬP X (setx):**

Gọi V(S) là tổng giá trị cách số trong tập S, nếu S={} thì V(S)=0.

Một tập được gọi là tập X nếu với mọi 2 tập con A, B khác nhau thì V(A) và V(B) cũng khác nhau, đồng thời các phần tử đều là số tự nhiên

VD: {3,4} là một tập X vì V({3,4}) <> V({3}) <> V({4}) <> V({})

Cho số tự nhiên N. hãy tìm V(X) nhỏ nhất có thể đạt được trong số các tập X có N phần tử

Chịu khó nhận xét toán

Inp: N<= 10^18

Out: V(X)min mod 10^9+7

|  |  |
| --- | --- |
| 1 | 1 |
| 2 | 3 |

**ĐÀO ĐƯỜNG 1D (dig1d):**

Tuyến đường bộ có 2+N ô liên tiếp đánh số từ 0 đến N+1. Mỗi điểm có độ an toàn là A[i]. A[0]=A[N+1]=0

Khi đứng tại ô thứ i, bạn chỉ có thể đi qua ô i+1 hoặc i+2. Độ an toàn của 1 đường đi từ 0 tới N+1 là tổng các độ an toàn trong các ô mà bạn đi qua.

Trước khi tiến hành đào đường tại ô thứ i người ta cần tính xem độ an toàn tối đa khi bắt buộc phải đi qua ô thứ i là bao nhiêu (ký hiệu X[i]), sau đó lại xác định độ an toàn tối đa khi cấm đi vào ô thứ i là bao nhiêu (ký hiệu Y[i])

Inp: N<=10^6. |A[i]|<=10^9

Out: gồm N dòng, dòng thứ i chứa 2 số nguyên X[i], Y[i]

Chào các bạn

**ĐÀO ĐƯỜNG 2D (dig2d):**

Một khu đất rộng MxN được chia thành đúng MxN ô vuông bằng nhau. tọa độ (x,y) chứa ô (x,y), ô này có độ an toàn là A[x,y]

Khi đứng tại ô (x,y), bạn chỉ có thể di chuyển đến ô (x+1,y) và (x,y+1). Độ an toàn của 1 đường đi từ ô (1,1) đến ô (M,N) được tính bằng cách lấy tổng tất cả độ an toàn trong các ô bạn đi qua

Khi tiến hành đào đường tại ô (x,y) họ cần xác định độ an toàn tối đa khi bắt buộc phải đi qua ô (x,y) là bao nhiêu (ký hiệu F[x,y]), và độ an toàn tối đa khi bị cấm đi vào ô (x,y) là bao nhiêu (ký hiệu G[x,y]). Ta coi như F[1,1]=G[1,1]=F[M,N]=G[M,N]=0

Chào các bạn 2D ?!?

Inp: M,N<=10^3. Ma trận A.

Out: ma trận F, G

**NÂNG NÚI (liftmount):**

Một dãy núi được chia thành N đoạn, đoạn thứ I có độ cao A[i]

Gọi phép lift(i) với 1<=I<N là phép tăng tất cả các đoạn từ 1🡪I lên 1 đơn vị độ cao đồng thời giảm từ i+1🡪n xuống 1 đơn vị độ cao

Tìm số phép lift ít nhất cần sử dụng để dãy núi có độ cao không tăng theo hướng 1🡪N

Đây là bài NĐTT huyền thoại, hướng nhìn lạ lẫm

Inp: N<=10^6; |a[i]|<=10^9

Out: số phép lift cần dùng

|  |  |
| --- | --- |
| 3  1 2 3 | 2 |

**CÁI ỐNG (pipe):**

Cho một cái ống đủ dài để chứa N viên sỏi, trên mỗi viên sỏi có số điểm A[i]. theo thứ tự, lần lượt bỏ các viên sỏi vào cuối ống. trong mọi thời điểm, ta có quyền bỏ sỏi ra từ 1 trong 2 đầu của ống.

Nếu viên sỏi i được lấy ra từ đầu ống, bạn được cộng x\*A[i] điểm

Nếu viên sỏi i được lấy ra từ cuối ống, bạn được cộng y\*A[i] điểm

Xác định số điểm lớn nhất có thể đạt được

Inp: N<=10^6; |x|,|y|<=10^6; |A[i]|<=10^6.

Out: số điểm lớn nhất có thể đạt được

|  |  |
| --- | --- |
| 5 3 2  -1 1 2 3 -3 | 10 |

Nó dễ lắm

**THU GỌN ĐỒ THỊ (curtailment):**

Cho một đồ thị vô hướng liên thông có N đỉnh và M cạnh, mỗi cạnh có trọng số C[i,j].

Hãy tìm cách giữ lại một số cạnh sao cho đồ thị này vẫn liên thông và tổng trọng số là nhỏ nhất có thể

Inp: N,M<=10^5; |C[i,j]|<=10^9

Out: tổng trọng số nhỏ nhất của các cạnh được giữ lại

How about cây khung?

**CỜ BẤT TẬN (titanchess):**

Quân cờ Titanchess có cách di chuyển giống như quân tượng trong cờ vua nhưng trong một lượt lại có thể đi tối đa 2 lần! đặc biệt: bàn cơ rộng bất tận!

Thế cờ hiện tại chỉ còn 2 quân cờ:

+ quân thứ nhất đứng tại ô (a,b);

+ quân thứ hai đứng tại ô (c,d);

Tại lượt đi của quân thứ nhất, hãy in ra:

+ 0 nếu không có cách nào quân 1 ăn được quân 2

+ 1 nếu chỉ cần 1 nước đi để ăn được quân thứ 2

+ 2 nếu cần đến 2 bước để ăn được quân thứ 2. Hãy cho biết tất cả tọa độ trung gian nó cần tới trước khi ăn quân thứ 2

Inp: T (số test case) <=10^6; |a|,|b|,|c|,|d|<=10^15.

Out: (như đề)

Đề dài chưa bao giờ là nỗi đáng sợ

**DÃY SỐ GATO (gseq):**

Cho dãy số nguyên A có N phần tử. một dãy gato là dãy con liên tiếp của A từ L đến R có đúng 1 cặp phần tử liên tiếp giống nhau

Hãy cho biết số lượng dãy gato trong A (1 cặp L, R khác nhau là 1 dãy gato khác)

Inp: N<=10^6; |A[i]|<=10^9;

Out: số lượng dãy gato trong A

|  |  |
| --- | --- |
| 5  1 2 3 3 2 | 6 |

Quy tắc đếm xàm thôi

**WIFI (wifi):**

Cho một mạng lưới gồm N+1 máy phát wifi được nối với nhau bởi M cổng thông tin 2 chiều. trong các máy có số hiệu từ 1🡪N mỗi máy có 2 thông số A[i] và R[i]

Khi tổng lượng thông tin đến máy i lớn hơn hoặc bằng R[i], chúng sẽ phát wifi đến những máy wifi có cổng thông tin nối với nó, mỗi máy thêm 1 lượng thông tin A[i]

Thiết bị phát đặc biệt có số hiệu 0. Được nối với 1 số máy trong N máy này. Thiết bị này có R[0]=0 nên không cần nhận thông tin. Hãy tính A[i] nhỏ nhất để tất cả các máy còn lại đều có thể nhận được wifi

Cũng khá quen thuộc

Inp: N,M<=10^5; đồ thị mô tả; 1<=A[i], R[i] <=10^9

Out: A[0] nhỏ nhất

**ĐĨA BAY (UFO):**

Đĩa bay ở tọa độ (x1,y1,z1) cần đến tọa độ (x2,y2,z2). biết rằng từ một ô bất kỳ chỉ được đi đến 1 trong 6 ô kề mặt với ô đó.

Đĩa bay cần di chuyển ít ô nhất. hãy tìm số cách để thực hiện điều đó

Inp: -10^5<=x1,y1,z1,x2,y2,z2<=10^5

Out: số cách di chuyển bằng ít ô nhất mod 10^9+7

|  |  |
| --- | --- |
| 2 2 2  3 3 3 | 6 |

hint: sử dụng toán đếm tổ hợp

**KHU GIẢI TRÍ (arcade):**

Có N máy trò chơi đút tiền. máy thứ i đang chứa A[i] và cần đạt được C[i] xu để tất cả xu rơi ra ngoài (gồm cả số tiền có sẵn trong máy và số tiền cho vào)

Trước khi vào khu trò chơi trên, bạn cần mua một số xu để tham gia chơi và chỉ mua đúng 1 lần. Hãy tìm số xu ít nhất cần mua để ăn hết tiền trong khu trò chơi

Inp: N<=10^6; 0<=a[i]<c[i]<=10^9

Out: số xu ít nhất cần dùng

|  |  |
| --- | --- |
| 3  1 2  1 3  1 4 | 1 |

Simple

**SỦA (bark):**

Có N chú chó đứng ở các tọa độ x[i], y[i].

Chú cho thứ i có thể sủa với độ lớn A[i] và nghe ở độ lớn B[i].

Nếu khoảng cách giữa chúng không quá D và A[i]>=B[j] thì con chó j có thể nghe thấy con chó i và nó cũng bắt đầu sủa.

Cho biết cần kích động ít nhất bao nhiêu con chó để tất cả N con chó đều sủa (ban đầu không có con chó nào sủa cả.

Inp: N<=1000; x[i],y[i],A[i],B[i] <= 10^9; D<=10^9

Out: số chó ít nhất cần kích động

Sủa, các bạn có thể đơn giản hóa đồ thị, chúc may mắn

**SINH MÃ (codegen):**

Mỗi bộ mã là 1 bộ 3 số (a,b,c), mỗi số có tối đa 2 chữ số.

Một phép sinh mã là một phép tạo ra một bộ mã mới từ bộ mã (a,b,c) như sau:

+ a’ = x\*(b+c);

+ b’ = y\*(a+c);

+ c’ = z\*(a+b);

Bằng cách lấy 2 chữ số cuối của 3 kết quả trên, ta được bộ 3 số mới!

Có Q câu hỏi, mỗi câu hỏi có 1 số K với ý nghĩa bạn cần cho biết nếu sinh mã K lần thì được bộ 3 số nào.

Inp: 0<=a,b,c<=100; Q<=10^6 truy vấn; mỗi truy vấn là 1 số K<=10^18

Out: Q câu trả lời tương ứng

Đây là dạng toán có chu trình (nhắc lại: sử dụng đồng dư, mod số phần tử trong chu trình)

**ĐOẠN LẶP LẠI (repeat):**

Cho dãy A có N phần tử. một dãy con tốt là 1 dãy con liến tiếp của A thỏa:

- có ít nhất 5 phần tử

- không có 2 phần tử liên tiếp nào giống nhau

Hãy tìm dãy con tốt có số lần lặp lại nhiều nhất (2 đoạn con được coi là khác nhau về vị trí khi điểm bắt đầu của chúng khác nhau)

Inp: N<=10^5; 0<=A[i]<=10^9;

Out: số lần lặp lại nhiều nhất trong số các dãy con tốt tìm được

Khó lắm sao :/

**BỎ CẦU (bridgecutting):**

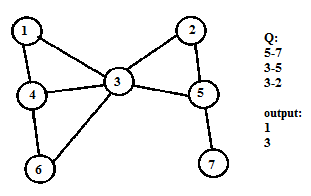
Cho một đồ thị vô hướng N đỉnh, M cạnh.

Cho 1 danh sách gồm Q cạnh, tại thời điểm thứ i, ta tiến hành bỏ cạnh thứ i trong danh sách trên (xóa luôn khỏi đồ thị, không gắn lại khi xóa cạnh sau đó).

Hãy cho biết các thời điểm mà ta cắt bỏ cầu.

Inp: N,M,Q<=10^5; danh sách Q;

Out: danh sách các thời điểm bỏ cầu



Bài này thì khó thiệt :D hãy thực hiện lại những gì đề yêu cầu

**MA TRẬN ƯỚC CHUNG LỚN NHẤT II (gcdmatII):**

Cho 2 mảng 1: chiều A có N phần tử và B có M phần tử

Gọi mảng 2 chiều G. trong đó G[x,y] = gcd(A[x], B[y])

Cho Q truy vấn có dạng x1, y1, x2, y2 hãy cho biết gcd của tất cả các số trong hình chữ nhật có đỉnh trái trên là (x1,y1) và phải dưới là (x2,y2) của mảng G

Inp: M,N,Q<=10^5; A[i], B[i]<=10^18; Q dòng thể hiện Q truy vấn trên

Out: Q dùng là đáp án cho Q truy vấn

Đây là cái gợi ý.

**QUÂN XÂM LƯỢC (invader):**

Qua một cổng không gian được mở tại tọa độ (N,M) và ở độ cao H, các tàu chiến của quân xâm lược bắt đầu ùa về xâm chiến Trái đất, chúng đáp cánh tại chung một tọa độ (1,1) ở độ cao 1.

Các tàu chiến không thay đổi tốc độ di chuyển và có tốc độ bằng nhau. Nếu giữ nguyên độ cao, từ ô (x,y) chúng chỉ có thể đi đến ô (x-1,y) hoặc ô (x,y-1). Bằng không, chúng có thể giữ nguyên tọa độ đồng thời giảm đi 1 đơn vị độ cao.

Biết được âm mưu của bọn xâm lược, hệ thống an ninh của trái đất đã chuẩn bị MxNxH đơn vị máy tính. Máy tính (x,y,z) theo dõi xem có bao nhiêu tàu chiến đi qua tọa độ (x,y) ở độ cao z.

Không may, bọn chúng đã phá hủy hệ thống máy tính này và tất cả những gì chúng ta biết được bây giờ là:

- có tàu chiến nào đi qua tọa độ (x,y) độ cao z hay không

- trong tất cả các số liệu thu được, vị trí có tàu chiến đi qua và có số lượng tàu ít nhất có K tàu chiến

Từ 2 dữ kiện trên hãy xác định số tàu ít nhất đã xâm chiến Trái Đất

Inp: N,M,H<=100; K<=10^9; H ma trận, ô (x,y) của ma trận thứ z mô tả rằng có tàu chiến đi qua ô (x,y) độ cao z hay không, nếu có là số 1, ngược lại số 0

Out: số tàu chiến ít nhất đã xâm lược

Đề lại dài rồi ☹ thử nghĩ cách làm trên phiên bản 2D và K=1 trước

**BỘI CHUNG NHỎ NHẤT (LCM):**

Cho 3 số tự nhiên A, B, K. hãy tìm đúng K bộ nghiệm nguyên cho phương trình sau:

Ax + By = LCM(A,B)

Inp: Q<=10^4 là số truy vấn; A,B <=10^8; K<=10;

Out: các nhóm dòng lần lượt là các bộ nghiệm của từng phương trình

Quen quá quá quen

**BƯỚC CHÂN (steps):**

Một cầu thang có N+2 bậc đánh số từ -1 đến N.

- Chân phải của bạn đang nằm tại bậc 0, và chân trái đang đặt vào bậc -1, bạn chỉ được bước 2 chân xen kẽ nhau (tức là cứ 1 lần chân trái, rồi đến phải, trái,… cứ như vậy).

- lần đặt chân tiếp theo phải đặt tại ô lớn hơn ô hiện tại.

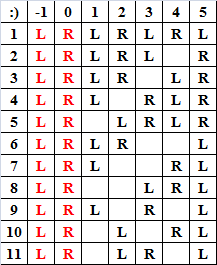
- khi bước, chân không được di chuyển quá 4 bậc thang

Hãy tìm số cách để có thể đặt 1 trong 2 chân đến ô thứ N.

Inp: N<=10^6

Out: số cách đặt chân đến ô N (mod 10^9+7)

|  |  |
| --- | --- |
| 5 | 11 |



Dễ thấy đây là 1 bài toán quy hoạch động :<