|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bài** | **Ghi chú** |
| 1 | Mạng điện - electronic |  |
| 2 | Tập X - setx |  |
| 3 | Cặp số ahihi 1 – ahihi1 |  |
| 4 | Chữa cháy – firefighter |  |
| 5 | Chữ số cuối cùng - lastdigit |  |
| 6 | Đào đường 1D – dig1d |  |
| 7 | Đào đường 2D – dig2d |  |
| 8 | Nâng núi - liftmount |  |
| 9 | Cái ống - pipe |  |
| 10 | Thu gọn đồ thị - curtailment |  |
| 11 | Sấm – thunder |  |
| 12 | Chỉ đường – theway |  |
| 13 | Rubik - rubik |  |
| 14 | Cờ bất tận - titanchess |  |
| 15 | Robot quay đầu – roboomerange |  |
| 16 | Cây tìm kiếm tam phân - TST |  |
| 17 | Cây tìm kiếm nhị phân - BST |  |
| 18 | Dãy số gato - gseq |  |
| 19 | Wifi - wifi |  |
| 20 | Cắt vùng – cutarea |  |
| 21 | Đĩa bay - UFO |  |
| 22 | Khu giải trí - arcade |  |
| 23 | Đánh số - numbering |  |
| 24 | Sủa – bark |  |
| 25 | Ma trận ước chung lớn nhất - gcdmatrix |  |
| 26 | Sinh mã – codegen |  |
| 27 | Tháp đều – righttower |  |
| 28 | Tổng fibonacci - fibonasum |  |
| 29 | Chọn kẹo - candy |  |
| 30 | Dãy đối xứng - reflectedseq |  |
| 31 | Vết sơn - stroke |  |
| 32 | Lặp lại - repeat |  |
| 33 | Ước fibonacci - divisionacci |  |
| 34 | Quân bài UNO - UNO |  |
| 35 | Đặt trạm radio - radio |  |
| 36 | Ước chung lớn nhất các cặp tích - ppgcd |  |
| 37 | Ma trận cấp số cộng – arithmatrix |  |
| 38 | Thành phần K liên thông - Komponent |  |
| 39 | Sinh chuỗi - stringen |  |
| 40 | Dòng sông – river |  |
| 41 | Bỏ cầu – bridgecuting |  |
| 42 | Dãy thăng trầm – updown |  |
| 43 | Tổng điểm số - hardscore |  |
| 44 | Vùng trung bình – equalarea |  |
| 45 | Trả tiền – money |  |
| 46 | Nén số - numzip |  |
| 47 | Bến xe – station |  |
| 48 | Điện trở - resistors |  |
| 49 | Cắm cờ - flags |  |
| 50 | Hái nấm – mushroom |  |

[QTN a tricky guy – 1.0]

**1. 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?

Inp:

- N<=10^6;

- M<=10^6;

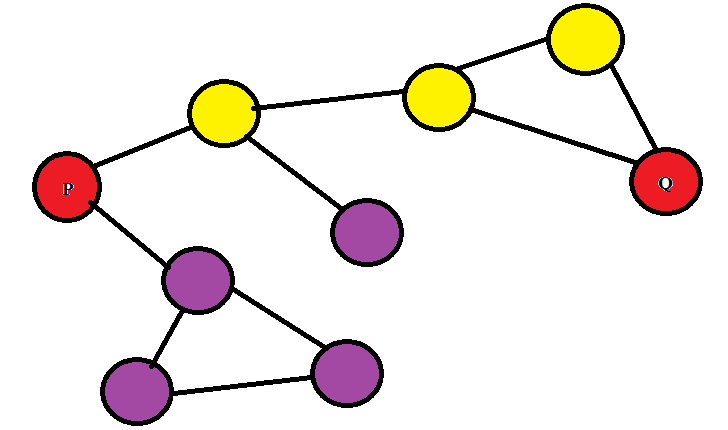
- P,Q<=N; P<>Q;

- 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



**2. 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

Inp: N<= 10^18

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

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

**3. CẶP SỐ AHIHI 1 (ahihi1):**

Cho 2 số x,y. 2 số a,b được gọi là cặp số ahihi nếu:

+ a<=b

+ a, b thuộc tập ước của x

+ UCLN(a,b)=y

Đếm số cặp số ahihi.

INP: x,y<=10^12

OUT: số lượng cặp số ahihi

|  |  |
| --- | --- |
| 8 4 | 2 |

Giải thích: 2 cặp (4;4), (4;8)

**4. CHỮA CHÁY (firefighter):**

Bạn đang đứng tại tọa độ (x1,y1), đồng thời tại tọa độ (x2,y2) đang xảy ra một vụ cháy.

Bước rằng dòng sông gần đó nhất là một đường thẳng song song với trục hoành đi qua điểm của tung độ y

Hãy tính toán quãng đường d ngắn nhất để bạn có thể lấy nước rồi đem tới ngôi nhà đó chữa cháy.

Inp: x1,y1,x2,y2,y <=10^18

Out: trunc(d)

|  |  |
| --- | --- |
| 3 4 9 4 0 | 10 |

**5. CHỮ SỐ CUỐI CÙNG (lastdigit):**

Một số rất lớn được tạo ra bằng cách viết liền các số từ X tới Y lại với nhau.

VD: X=13; Y=15 thì số đó sẽ là 131415

Gợi 1 phép UP là phép:

+ lấy 2 chữ số cuối ra khỏi số hiện tại

+ lẩy tổng 2 chữ số đó gắn ngược lại lên đầu.

Sau một số lần hữu hạn phép UP, con số chúng ta nhận được chỉ còn 1 chữ số. hãy xác định chữ số đó.

Số test case: T

Inp: T<=10^6, X, Y<=10^18

Out: T dòng cho biết chữ số cuối cùng của từng test case trong input

|  |  |
| --- | --- |
| 1  13 15 | 6 |

Giải thích: số tạo được là 131415 🡪 61314 🡪 5613 🡪 456 🡪 114 🡪 51 🡪 6

**6. 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]

**7. 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

Inp: M,N<=10^3. Ma trận A.

Out: ma trận F, G

**8. 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

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

Out: số phép lift cần dùng

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

**9. 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 |

**10. 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

**11. SẤM (thunder):**

Có N người đứng theo thứ tự, xếp thành hàng ngang, người thứ i có độ cao A[i]

Biết rằng sấm sẽ đánh vào người thứ i nếu người này không thấp hơn cả 2 người bên cạnh

Sau khi 1 người bị đánh chết, những người còn lại sẽ dồn vào và tiếp tục quá trình trên cho đến khi không còn ai có thể bị đánh nữa.

Xác định chỉ số của những người còn lại

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

Out: K số người còn sống sót. Và chỉ số K người đó

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

**12. CHỈ ĐƯỜNG (theway):**

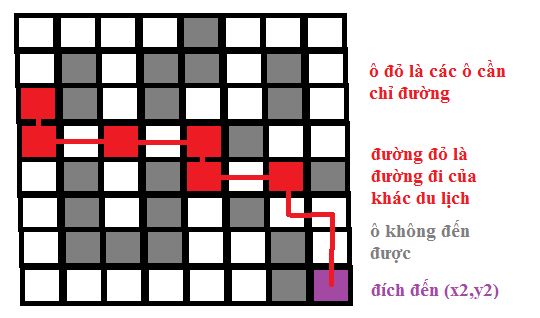
Bạn cần phải chỉ đường cho 1 người du lịch. Người này đang đứng tại ô (x1,y1) và cần di chuyển đến ô (x2,y2). Các ô có thể tới được kí hiệu bởi số 0, ngược lại ký hiệu số 1.

Người du lịch chỉ có thể di chuyển đến 1 trong 4 ô kề cạnh ô anh ta đang đứng.

Anh ta luôn tự hiểu rằng không cần phải quay lại những ô đã đi qua. Tại những nơi có nhiều hơn 1 hướng đi, bạn cần chỉ cho anh ta hướng đi thích hợp

Inp: M,N<=10^3; mảng 2 chiều thể hiện map; x1,y1,x2,y2<=M,N

Out: số lần chỉ đường ít nhất, nếu có nhiều cách chỉ như vậy, hãy cho biết độ dài ngắn nhất mà người du lịch phải đi trong những cách chỉ ít nhất đó



(hình ảnh chỉ mang tính minh họa, đường đi trên không thể hiện kết quả test tương ứng)

**13. RUBIK (rubik):**

Để vẽ lại bầu trời đêm, người ta chia bầu trời thành một lưới ô vuông gồm 3N cột và 3M hàng.

Người ta sử dụng các khối Rubik 3x3x3, có mặt đen và mặt vàng đối diện nhau. mỗi khối Rubik sẽ thể hiện 1 ô vuông 3x3 trong mạng lưới, màu vàng cho nơi có sao và màu đen cho nơi không có sao.

Biết rằng bạn chỉ được phép xoay 1 cạnh ra phía sau. Hãy tìm số lần quay Rubik ít nhất để vẽ lại được bầu trời đó, ban đầu các khối Rubik đều đã được giải.

Inp: N,M<=1000; ma trận A[1..3N,1..3M] thể hiện bầu trời đó

Out: số lần quay Rubik ít nhất, nếu không có cách nào thì in 0.

**14. 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)

**15. ROBOT QUAY ĐẦU (roboomerange):**

Cho 1 dãy N ô, mỗi ô chứa 1 giá trị A[i]. Roboomerange (viết tắt Rb) được lập trình cho trò chơi như sau:

+ Rb có 1 lượt đi từ 1🡪N và ngược lại N🡪1

+ mỗi ô đi qua, Rb có quyền chọn hoặc bỏ qua ô đó

+ ô chọn sau phải có giá trị lớn hơn giá trị ô được chọn trước đó

+ khi chọn được 1 ô, Rb được cộng 1 điểm

Hãy xác định số điểm tối đa mà Rb có thể đạt được trong trò chơi trên

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

Out: số điểm tối đa

**16. CÂY TÌM KIẾM TAM PHÂN (TST):**

Cây tìm kiếm tam phân (Ternary Search Tree – TST) là cấu trúc dữ liệu chế ra cho đề bài này!

Cây là cây tam phân, có đúng 1 nút gốc. mỗi nút cha có tối đa 3 nút con. Mỗi nút chứa 2 giá trị (x[i], y[i]), biết x[i]<=y[i]. và với nút i bất kỳ không phải lá:

+ mọi con cháu của nó về phía bên trái đều có giá trị bé hơn x[i]

+ mọi con cháu của nó về phía bên phải đều có giá trị lớn hơn y[i]

+mọi con cháu ở giữa đều có giá trị lớn hơn x[i] nhưng nhỏ hơn y[i]

Cho 1 dãy số N phần tử có các giá trị lần lượt là 1..N, hãy tìm số cách biểu diễn N phần tử này thành 1 TST

Inp: N<=100

Out: số cách biểu diễn mod cho 10^9+7

|  |  |
| --- | --- |
| 3 | 10 |

**17. CÂY TÌM KIẾM NHỊ PHÂN (BST):**

Cho dãy số có N phần tử có giá trị lần lượt là 1..N. có tất cả bao nhiêu cây tìm kiếm nhị phân có thể biểu diễn được dãy số đó?

Inp: N<=10^5

Out: số cách biểu diễn dãy số đó

|  |  |
| --- | --- |
| 4 | 14 |

**18. 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 |

**19. 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

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

Out: A[0] nhỏ nhất

**20. CẮT VÙNG (cutarea):**

Cho N lần vẽ, mỗi lần phải vẽ M đường thẳng song song với nhau.

Hãy cho biết số vùng tối đa tạo được bởi M.N đường thẳng đó

Inp: 0<=N,M<=10^18

Out: số vùng tối đa tạo được mod 10^9 +7

|  |  |
| --- | --- |
| 2 2 | 9 |

**21. 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 |

**22. 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 |

**23. ĐÁNH SỐ (numbering):**

Có bao nhiêu cách điền vào bảng 2 hàng, N cột theo quy tắc sau:

- số 1 có thể đặt vào bất kỳ đâu trong bảng

- mọi số i (1 < i <= 2N) đều phải kề cạnh với số i-1

Inp: N<=10^18

Out: số cách điền mod 10^9 + 7

|  |  |
| --- | --- |
| 2 | 4 |

**24. 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

**25. MA TRẬN ƯỚC CHUNG LỚN NHẤT (gcdmatrix):**

Cho biết ma trận:

- kích thước M.N (M hàng, N cột);

- Cho trước hàng đầu và cột đầu

- A[i,j] = gcd( A[i-1,j], A[i,j-1] )

Cho Q truy vấn dạng x,y. yêu cầu xuất ra giá trị ô (x,y);

Inp: M,N,Q<=10^5; A[1..i,1], A[i,1..i]<=10^18. Các truy vấn gồn 2 số x,y <=10^5

Out: kết quả tương ứng Q truy vấn

**26. 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.

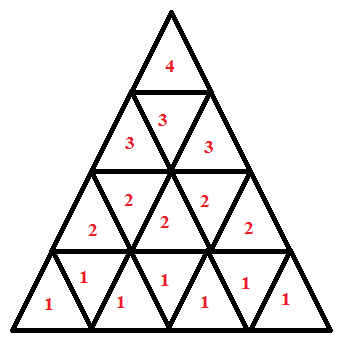
Inp: 0<=a,b,c<=100; Q<=10^6 truy vấn; mỗi truy vấn là 1 số K<=10^18

Out: Q câu trả lời tương ứng

**27. THÁP ĐỀU (righttower):**

Các viên gách hình tam giác đều có độ cao là 1.

Trên mặt phẳng, người ta dựng 1 tháp cũng hình tam giác đều với độ cao N (xem hình). Và như vậy người ta có thể chia tháp thành N tầng bằng nhau (mỗi tầng cao 1). Các tầng tại tầng thứ i (từ dưới đến lên) sẽ mang giá trị i.

 Hãy tính số viên gạch cần sử dụng và tổng giá trị trên tất cả các viên gạch

Inp: N<=10^18

Out: kết quả như đề yêu cầu mod 10^9 + 7

Hình bên minh họa trường hợp N=4

Khi đó

+ số viên gạch cần sử dụng là 16

+ Tổng giá trị trên các viên gạch là 30

**28. TỔNG FIBONACCI (fibonasum):**

Cho 2 số fibonacci thứ X và thứ X+1. Hãy tính tổng tất cả các số fibonacci từ L đến R.

Inp: X<=10^18; L<=R<=10^18; F[X], F[X+1]<=10^9;

Out: kết quả mod 10^9 +7

**29. CHỌN KẸO (candy):**

Cho N hộp kẹo, hộp thứ i có i viên kẹo.

Cần chọn ra K viên kẹo từ cùng 1 hộp bất kỳ.

Hãy cho biết số cách chọn

Inp: N,K<=10^5;

Out: số cách chọn mod 10^9;

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

**30. DÃY ĐỐI XỨNG (reflectedseq):**

Dãy A được định nghĩa như sau:

+ A[1]; A[0] cho sẵn

+ A[i]=A[i-1]+A[i-2]+A[i-3]; với mọi i>=2;

+ A[i]=A[-i] với mọi i

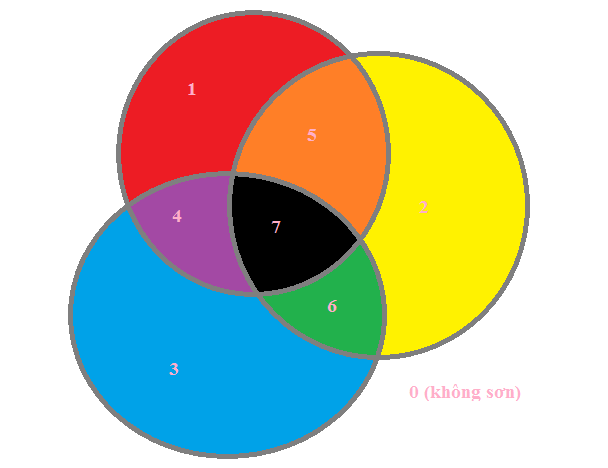
Hãy tính tổng các phần tử từ L đến R trong dãy A.

Inp: 0<=A[0], A[1]<=10^9; -10^18<=L<R<=10^18

Out: lấy 9 chữ số cuối của tổng tìm được.

**31. VẾT SƠN (stroke):**

Một nền nhà rộng MxN, mỗi ô có thể có hoặc không được sơn màu. ồ C[i,j] biểu diễn màu của ô (i,j) với quy ước:



Quy tắc phối màu:

+ 0+x⬄x với mọi x

+ 1+2⬄5

+ 1+3⬄4

+ 2+3⬄6

+ 1+2+3⬄7

Theo bảng trên màu tím đi chung với màu vàng vẫn cho ra màu đen vì:

4+2⬄1+3+2⬄7

Đứng từ ô (1,1) của nền nhà trên, cần đi đến ô (M,N) mà chỉ được sử dụng bước qua phải và đi xuống. màu của hành trình là sự phối màu của tất cả các ô đã đi qua. Hãy tìm số cách đi đến đích với từng loại màu

Inp: M,N<=10^6; ma trận C;

Out: 8 số nguyên, số thứ i cho biết số cách di chuyển đến ô (M,N) mà hành trình mang màu i là bao nhiêu cách (mod 10^9 + 7)

**32. 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

**33. ƯỚC FIBONACCI (divisionacci):**

Dãy Fibonacci là dãy:

+ F[1]=F[2]=1;

+ F[i] = F[i-1]+F[i-2];

Cho N, hãy cho biết F[N] chia hết cho tất cả bao nhiêu số fibonacci **khác**

Inp: N<=10^12;

Out: số lượng số fibonacci là ước của F[N] ngoại trừ chính nó.

|  |  |
| --- | --- |
| 4 | 2 |

**34. QUÂN BÀI UNO (uno):**

Trong bộ bài UNO, mỗi lá bài mang trên mình một màu và một con số.

Biết bộ bài này có N lá bài, các số trên bài là các số nguyên dương không quá A, và có B màu đánh số từ 1 tới B.

Quân X có thể nối tiếp quân bài Y nếu X cùng màu với Y hoặc X cùng số với Y

Với một số lá bài được chọn, hãy cho biết có hay không tồn tại một cách sắp xếp tất cả các lá bài thành 1 chuỗi dài ở đó mỗi lá bài (trừ lá đầu tiên) đều nối tiếp lá trước đó của nó.

Inp: N<=10^6; A\*B<=10^6; N dòng cho biết số và màu trên các quân bài

Out: số 1 tương ứng với việc có tồn tại cách sắp xếp như vậy, ngược lại in ra số 0

**35. ĐẶT TRẠM RADIO (radio):**

Có 3 đối tượng cần thông tin liên lạc, đó là 2 trạm M, N trên mặt phẳng tọa độ và đường băng (d): ax+by+c=0. Hãy xác định chi phí vận hành nhỏ nhất theo nội dung sau:

- ta có thể đặt trạm máy chủ ở bất kì đâu trên mặt phẳng tọa độ, gọi điểm đặt trạm máy chủ là O.

- nếu từ O tồn tại 1 giá trị R sao cho đường tròn tâm O bán kính R có đi qua A, B và đường thẳng (d), ta đã tìm ra 1 cách lắp đặt với chi phí là [R] là giá trị nguyên làm tròn lên của R.

Inp: 1<= xM, yM, xN, yN <=10^9; 0<a, b, c<=10^9

Out: [R] nhỏ nhất có thể đạt được

**36. ƯỚC CHUNG LỚN NHẤT CÁC CẶP TÍCH (ppgcd):**

Cho dãy A gồm N phẩn tử. dãy A’ là tập hợp các tích của 2 phần tử khác nhau của A.

A’ = { A[i]\*A[j] với mọi i<>j; 1<=i,j<=N }

Hãy tính ước chung lớn nhất của tất cả các phần tử trong mảng A’

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

Out: ước chung lớn nhất của tất cả các phần tử trong A’

**37. MA TRẬN CẤP SỐ CỘNG (arithmatrix):**

1 cấp số cộng có công sai d và số hạng đầu U[1] được định nghĩa như sau:

U[i] = U[i-1] + d với mọi i>1;

Gọi mảng 2 chiều S được định nghĩa như sau:

S[1,i] = U[i];

S[i,1] = U[1];

S[i,j] = S[i-1,j] + S[i,j-1]; với mọi (i,j>1)

Hãy xác định S[x,y]

Inp: U[1], d<=10^9; x,y<=10^5;

Out: S[x,y] mod 10^9+7

**38. THÀNH PHẦN K\_LIÊN THÔNG (komponent):**

Một thành phần K liên thông là 1 thành phần liên thông mà ở đó, khi bỏ K-1 cạnh bất kỳ, đồ thị vẫn liên thông

Cho một đồ thị vô hướng liên thông, biết rằng đây là 1 thành phần K liên thông. hãy xác định số K lớn nhất.

Inp: N<=100; M<=10^4;

Out: K

**39. SINH CHUỖI (stringen):**

Xếp các ký tự từ “A” đến “Z” thành 1 vòng tròn, “A” đứng trước “B”, “B” trước “C”, …, “Z” trước “A”.

Ban đầu chuỗi chỉ gồm 1 ký tự. gọi 1 phép stringen là thao tác sau:

- chọn các ký tự nằm ở vị trí lẻ

- đối với 1 ký tự được chọn: xét trên vòng tròn, từ ký tự đó qua bên trái L bước được ký tự Cl, và đếm qua bên phải R bước được ký tự Cr. Chèn vào trước chữ cái đang xét ký tự Cl và sau chữ cái đó chèn Cr

Vd:

L=1; R=2;

‘G’ 🡪 ‘FGI’ 🡪 ‘EFHGHIK’ 🡪 …  
 cho trước 1 dãy gồm Q hoạt động. hoạt động loại 1 là stringen, hoạt động loại 2 kèm với 1 số x là yêu cầu xác định ký tự thứ x trong dãy.

Inp: chữ cái ban đầu; Q<=10^5; L,R<=10^9; Q dòng mỗi dòng là số 1 hoặc bộ số 2 và X tương ứng như trên biết rằng tổng số yêu cầu loại 1 không quá 60.

Out: với mỗi yêu cầu loại 2, xuất ra ký tự tương ứng

**40. DÒNG SÔNG (river):**

Một dòng sông có chiều dài L, và chiều rộng W được chia thành bảng LxW ô, W hàng, L cột.

Trên sông có N tảng đá lớn ở các ô (X[i],Y[i]).

Bạn có thể đặt thuyền tại bất kỳ ô nào trên cột đầu tiên miễn sao đó không phải là đá.

Bạn không thể di chuyển ngược dòng, chỉ có thể di chuyển lên, xuống hoặc qua phải, không được đi lại ô đã đi và không được đi vào đá.

Hãy cho biết số cách di chuyển lên bờ bên kia tại cuối nguồn (bờ sau cột cuối cùng), tất nhiên bạn có thể lên bờ tại bất kỳ hàng nào

Inp: L<=10^9, R<=1000; N<=1000; N dòng cho biết tọa độ các tảng đá; 1<=X[i]<=L; 1<=Y[i]<=W

Out: số cách lên bờ bên kia

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

**41. BỎ CẦU (bridgecutting):**

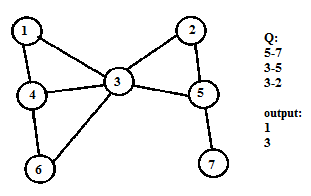
Cho một đồ thị vô hướng N đỉnh, M cạnh.

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



**42. DÃY THĂNG TRẦM (updown):**

Dãy A cần thỏa các tính chất sau:

- N phần tử

- các phần tử chỉ mang 3 loại giá trị 1, 2, 3

- đảm bảo rằng với mọi 1<i<N thì (A[i-1]-A[i])\*(A[i+1]-A[i]) > 0

Hãy cho biết có bao nhiêu dãy như vậy

Inp: N<=10^18;

Out: số dãy A thỏa đề

|  |  |
| --- | --- |
| 3 | 10 |

1 2 1

1 3 1

1 3 2

2 1 2

2 1 3

2 3 1

2 3 2

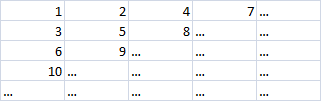
3 1 2

3 1 3

3 2 3

**43. TỔNG ĐIỂM SỐ (hardscore):**

Với 1 mảng 2 chiều rộng vô cùng, lần lượt điền các số nguyên dương theo thứ tự:



Ban đầu đứng tại ô (1,1), và chỉ được di chuyển theo 2 hướng: xuống hoặc sang phải.

Sau khi bước đủ N bước, người ta sẽ tính điểm của đường đi bằng cách lấy tổng số ghi trên các ô đã đi qua.

Hãy tính tổng tất cả các số điểm có thể đạt được sau N bước đi

Inp: N<=10^18

Out: tổng tất cả điểm mod 10^9 + 7

|  |  |
| --- | --- |
| 2 | 34 |

1+2+4=7

1+2+5=8

1+3+5=9

1+3+6=10

**44. VÙNG TRUNG BÌNH (equalarea):**

Xét một vùng G mặt phẳng tọa độ [1..M,1..N]

Nếu 1 ô có khoảng cách mahattan đến ô (x1,y1) bằng khoảng các mahattan đến ô (x2,y2), ô đó gọi là ô trung bình.

Vùng trung bình của 2 ô (x1,y1) và (x2,y2) là tất cả các ô trung bình của 2 ô này nằm trong vùng G.

Hãy cho biết vùng trung bình có bao nhiêu ô?

Inp: Q<=10^6 truy vấn, mỗi truy vấn là 6 số 1<=M,N,x1,y1,x2,y2<=10^9;

Out: Q dòng cho các câu trả lời tương ứng

**45. TRẢ TIỀN (money):**

Có 4 loại tiền mang giá trị A, B, C, D. có bao nhiêu cách trả tiền một món hàng N mà không dư tờ tiền nào? (1 tờ tiền được coi là dư khi bỏ tờ tiền đó ra vẫn mua được món hàng trên)

Inp: N<=10^5; 0<A<B<C<D<10

Out: số cách mod 10^9 + 7

|  |  |
| --- | --- |
| 4 1 2 5 10 | 5 |

1 1 1 1

1 1 2

2 2

5

10

**46. NÉN SỐ (numzip):**

Cho 1 mảng 2 chiều M\*N, mỗi ô (x,y) có giá trị A[x,y]. hãy tìm cách tối thiểu hóa các số trên bảng sao cho:

- 2 ô kề nhau (có ít nhất 1 điểm chung) vẫn giữ nguyên trạng thái lớn hơn, bé hơn, bằng.

- tổng tất cả các ô trong bảng là tối thiểu

Hãy cho biết tổng các ô trong bảng nén tối thiểu là bao nhiêu?

Inp: M,N thỏa M\*N<=10^5. A[x,y]<=10^9

Out: tổng tối thiểu

|  |  |
| --- | --- |
| 4 3  1 1 1 9  1 2 2 2  8 8 7 7 | 27 |

1 1 1 3

1 2 2 2

4 4 3 3

**47. BÊN XE (station):**

1 bến xe gồm N khu vực để xe xếp thành hàng ngang và theo thứ tự đánh số từ 1 đến N.

Để nâng cấp tiện ích trao đổi thông tin, người ta cần xây lên đúng 2 cột ăng-ten. Ta có thể dựng 1 ăng-ten giữa khu vực đỗ xe thứ i và khu vực đỗ xe thứ i+1 với mọi 1<=i<N.

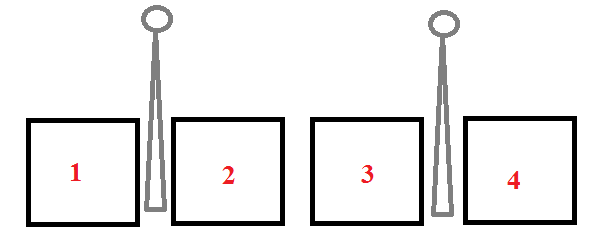
2 khu vực để xe có thể truyền thông tin cho nhau nếu như giữa chúng có đúng 1 cột ăng-ten. (nếu qua 2 cột ăng-ten việc liên lạc có thể bị nhiễu sóng)

Hãy cho biết có thể có nhiều nhất bao nhiêu cặp số (i,j) thỏa i<j và khu vực thứ i có thể liên lạc đến khu vực thứ j qua ăng ten. đồng thời cho biết có bao nhiêu cách lắp đặt ăng-ten để đạt được số cặp (i,j) tối đa trên

Inp: 2<N<=10^18

Out: số cặp xe nhiều nhất, số cách lặp đặt thỏa yêu cầu đề bài

|  |  |
| --- | --- |
| 4 | 4  1 |



Nếu lắp ở các vị trí như trong hình, có 4 cặp xe có thể liên lạc với nhau:

1-2;

1-3;

2-4;

3-4

không có cách lắp đặt khác cũng cho 4 cặp xe như vậy

**48. ĐIỆN TRỞ (resistors):**

Có N bóng đèn nối với nhau bởi M dây dẫn. bóng đèn i có điện trở Rl[i]. dây nối từ đèn i đến j có điện trở Rw[i,j]

Để trang trí, hãy tìm một chu trình đơn có chứa đèn thứ x và đèn thứ y sao cho tổng các điện trở trên đèn và dây là nhỏ nhất có thể

Inp: 1<=x,y<=N<=100; M<=10000; 0<Rl[i]<=10^9; 0<Rw[i,j]<=10^9

Out: tổng điện trở nhỏ nhất có thể đạt được

**49. CẮM CỜ (flags):**

Thành phố có tổng cộng N khu phố, đồng thời có M đường nối 2 chiều giữa một số khu phố. Ta coi như thời gian di chuyển trên mỗi đoạn đường là như nhau. bạn tham gia 1 trò chơi như sau:

- ban tổ chức sẽ cho bạn 2 địa điểm s, t là mã số của 2 khu phố bắt đầu và đích đến

- bạn được giao K lỗ cắm. bạn cần bố trí ít nhất 1 lỗ ở khu phố s, ít nhất 1 ổ ở khu phố t, các ổ còn lại bạn có thể đặt tùy thích ở các khu phố khác. Tại 1 khu phố bất kỳ, các lỗ cắm sau khi bố trí vào sẽ được đánh số thứ tự

- sau khi bố trí tất cả lỗ cắm, bạn được giao 3 lá cờ. nhiệm vụ của bạn là cắm lá cờ đầu tiên vào 1 lỗ cắm ở khu phố s, lá cờ thứ 2 cắm vào 1 lỗ cắm thuộc khu phố bất kỳ có lỗ cắm dọc trên đường đi, lá cờ thứ 3 cắm vào 1 lỗ cắm ở khu phố t.

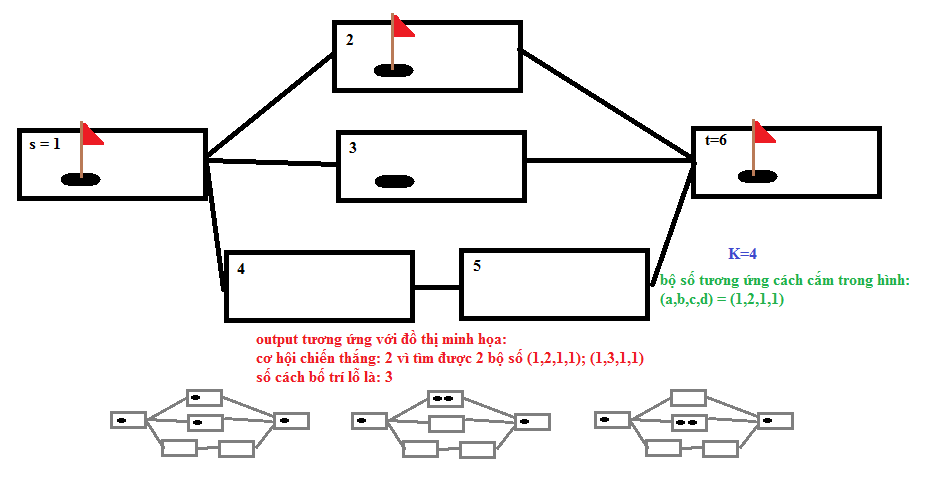
- hiển nhiên để nhiệm vụ hoàn thành xong nhanh nhất có thể, bạn sẽ chọn 1 trong số những con đường ngắn nhất từ s tới t.

- gọi a là chỉ số lỗ cắm tại vị trí s bạn đã cắm, b là khu phố bạn cắm lá cờ thứ 2, c là chỉ số lỗ cắm tại thành phố b mà bạn đã cắm lá cờ thứ 2 và d là chỉ số lỗ cắm tại vị trí t bạn đã cắm.

Gọi cơ hội chiến thắng là số lượng bộ 4 số (a,b,c,d) có thể tìm được, hãy tìm cơ hội chiến thắng lớn nhất, đồng thời cho biết có bao nhiêu cách bố trí lỗ cắm để có cơ hội chiến thắng lớn nhất đó

Inp: N, M, K<=10^5; s,t<=N; danh sách cạnh

Out: 2 số nguyên cho biết cơ hội chiến thắng lớn nhất và cách bố trí lỗ cắm, cả 2 con số này đều lấy đồng dư cho 10^9+7



Cách bố trí thứ 2: (1,2,1,1) và (1,2,2,1). Cách bố trí thứ 3: (1,3,1,1) và (1,3,2,1)

**50. HÁI NẤM (mushroom):**

Cho một khu rừng hình chữ nhật rộng N\*M (N cột, M hàng).

Một cậu bé có nhiệm vụ đi từ ô (1,1) đến ô (N,M) để hái nấm, lúc đi cậu chỉ được đi 2 hướng: xuống hoặc qua phải.

Lúc về cậu chỉ được đi 2 hướng còn lại: lên trên và qua trái. Vì tính cách nghịch ngợm, cậu sẽ không đi lại y hệt con đường lúc đi mà sẽ rẽ một hướng mới tại 1 ô nào đó. Dù vậy cậu vẫn sợ bị lạc, nên trong lúc đang đi hướng mới, một khi đã bước lại 1 ô trên con đường lúc cậu đi, cậu sẽ chỉ đi theo con đường đó về nhà mà không rẽ theo hướng nào khác nữa!

Hãy cho biết số cách mà cậu bé có thể hoàn thành nhiệm vụ trong 1 lượt đi – về

Inp: N,M<=200

Out: số cách hoàn thành nhiệm vụ mod 10^9+7

|  |  |
| --- | --- |
| 3 3 | 26 |

