

06/11:

Bài A. MTC

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

Dãy $X = x_1, x_2, \dots, x_k$ được gọi là núi nếu tồn tại số tự nhiên i sao cho:

- $x_j < x_{j+1} \quad \forall 1 \leq j < i$
- $x_j > x_{j+1} \quad \forall i \leq j < k$

Cho dãy số nguyên a , hãy đếm số cách xóa đi một số phần tử của a sao cho dãy còn lại là núi. Hai cách xóa được cho là khác nhau nếu tồn tại một vị trí được xóa trong cách này nhưng không được xóa trong cách kia.

Dữ liệu vào

- Dòng đầu chứa số phần tử của dãy a : n
- Dòng tiếp theo chứa dãy a

Kết quả

In ra phần dư của số cách xóa khi chia cho $10^9 + 7$

Ví dụ

stdin	stdout
5 1 2 3 2 1	27

Giải thích

Các cách xóa thỏa mãn là (dãy con sau khi xóa): \emptyset , 1, 2, 3, 2, 1, 12, 13, 12, 23, 21, 32, 31, 21, 123, 121, 132, 131, 121, 232, 231, 321, 1232, 1231, 1321, 2321, 12321

Hạn chế

- $n \leq 10^5$. $-10^9 \leq a_i \leq 10^9$
- 30% test với $n \leq 20$
- 30% test với $20 < n \leq 1000$

Bài B. KDQUEEN

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

Cho một bàn cờ k chiều kích thước $n_1 \times n_2 \times \dots \times n_k$. Một quân hậu được đặt ở vị trí $X = (x_1, x_2, \dots, x_k)$, nó có thể di chuyển sang vị trí $Y = (y_1, y_2, \dots, y_k)$ ($1 \leq y_i \leq n_i$) nếu X và Y cùng hàng hoặc cùng đường chéo. Cùng hàng nghĩa là tồn tại i sao cho $|x_i \neq y_i|$ và $x_j = y_j \forall j \neq i$. Cùng đường chéo nghĩa là $|x_i - y_i| = |x_j - y_j| \forall 1 \leq i < j \leq k$. Hãy đếm số ô mà quân hậu có thể di chuyển sang (trong một nước đi).

Dữ liệu vào

- Dòng đầu ghi số nguyên dương k là số chiều của bàn cờ;
- Dòng thứ hai ghi k số nguyên dương n_1, n_2, \dots, n_k là kích thước của bàn cờ;
- Dòng thứ ba ghi k số nguyên dương x_1, x_2, \dots, x_k là vị trí của quân hậu.

Kết quả

Ghi số ô mà quân hậu có thể di chuyển sang sau khi chia lấy dư cho $10^9 + 7$

Ví dụ

stdin	stdout
3 3 3 3 1 2 3	8

Giải thích

Các ô đó là $(3, 2, 3); (2, 2, 3); (1, 1, 3); (1, 2, 1); (1, 3, 3); (1, 2, 2); (2, 1, 2); (2, 3, 2)$.

Hạn chế

- Trong tất cả các test: $2 \leq k \leq 10^5$, $2 \leq n_i \leq 10^9$;
- Có 20% số test với $k \leq 5$ và $n_i \leq 100$;
- Có 20% số test với $k \leq 1000$ và $n_i \leq 1000$;
- Có 20% số test với $n_i \leq 10^5$;
- Có 40% số test với ràng buộc gốc.

Bài C. FTREE

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

Đất nước "vui vẻ" có n thành phố được đánh số từ 1 đến n , và $n - 1$ con đường hai chiều nối một số cặp thành phố. Giữa hai thành phố bất kỳ luôn có đường đi trực tiếp hoặc đường đi gián tiếp qua một số thành phố trung gian

Có k thành phố nổi tiếng mà khách du lịch luôn muốn đến thăm. Khách du lịch sẽ bắt đầu từ một thành phố nào đó trong n thành phố, đi theo các con đường nối giữa các thành phố để thăm đủ k thành phố nổi tiếng (mỗi con đường/thành phố có thể đi qua nhiều lần). Đất nước khá là vui vẻ trừ giá taxi ở đây, nên khách du lịch muốn chọn lộ trình có tổng độ dài phải đi là nhỏ nhất. Nhưng bạn biết đấy, sẽ thật phiền phức khi vừa đi du lịch vừa phải code, họ sẽ rất vui vẻ nếu nhận được sự giúp đỡ của bạn

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương n, k
- $n - 1$ dòng tiếp theo, mỗi dòng chứa 3 số nguyên dương u, v, L ; cho biết có một con đường độ dài L nối giữa hai thành phố u, v
- Dòng tiếp theo chứa k số nguyên dương là số hiệu của k thành phố nổi tiếng

Kết quả

Gồm n dòng, dòng thứ i là tổng độ dài phải đi của lộ trình tốt nhất nếu khách du lịch bắt đầu từ thành phố i

Ví dụ

stdin	stdout
13 5	13
1 2 1	12
2 3 1	11
3 4 1	10
2 5 1	13
1 7 1	13
6 7 1	12
7 10 1	12
1 8 1	11
8 9 1	13
8 11 1	13
8 12 1	11
12 13 1	10
2 4 7 9 13	

Hạn chế

- $1 \leq k \leq n \leq 10^5$. $1 \leq L \leq 10^9$
- Có 50% số test với $n \leq 1000$

Bài D. PCOUNT

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

Với $p = p_1, p_2, \dots, p_n$ là một hoán vị của n số nguyên dương từ 1 đến n , gọi $\beta(p)$ là số vị trí i có $p_i < i$. Cho n và k , hãy đếm số hoán vị p của n số từ 1 đến n có $\beta(p) = k$

Dữ liệu vào

- Một dòng duy nhất chứa n k

Kết quả

- Ghi số lượng hoán vị tìm được sau khi chia lấy dư cho $10^9 + 7$

Ví dụ

stdin	stdout
4 1	11

Hạn chế

- $1 \leq n \leq 1000$, $0 \leq k \leq n$
- Có 30% số test với $1 \leq n \leq 9$
- Có 30% số test với $9 < n \leq 20$

Bài E. MEGARR

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

Cho 2 mảng số nguyên a_1, a_2, \dots, a_n và b_1, b_2, \dots, b_m . Mảng c được xây dựng bằng cách lấy tất cả tích $a_i \times b_j$ của mọi cặp (i, j) , $1 \leq i \leq n$, $1 \leq j \leq m$. Tìm số bé thứ k trong dãy c (số bé nhất được hiểu là số bé thứ 1)

Dữ liệu vào

- Dòng đầu chứa số k ($1 \leq k \leq n \times m$)
- Dòng thứ 2 chứa số n và n số nguyên a_1, a_2, \dots, a_n
- Dòng cuối cùng chứa số m và m số nguyên b_1, b_2, \dots, b_m

Kết quả

- Một số nguyên duy nhất là kết quả bài toán

Ví dụ

stdin	stdout
6 3 1 2 3 2 0 -1	0

Hạn chế

- $1 \leq n, m \leq 10^5$, $|a_i|, |b_i| \leq 10^9$
- 50% số test có $1 \leq n, m \leq 5000$

Bài F. FBRA

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

Từ thời xa xưa, dãy ngoặc được xem như là biểu tượng của tình yêu đôi lứa. Với Hoang_Thuy, nó còn là lần đầu anh gặp Thuy_Hoang. Một cách tự nhiên như cách con người tìm ra lửa, anh ta trở nên yêu quý các dãy ngoặc đúng:

- Xâu rỗng là một dãy ngoặc đúng
- Nếu A là dãy ngoặc đúng thì $(A), [A], \{A\}$ cũng là dãy ngoặc đúng
- Nếu A, B là 2 dãy ngoặc đúng thì AB cũng là dãy ngoặc đúng

Mọi việc đều êm đẹp cho đến một ngày, những dãy ngoặc có vẻ đúng của Hoang_Thuy viết tặng cho Thuy_Hoang bị mờ đi một vài chỗ. Bằng cách điền các dấu ngoặc vào những chỗ bị mờ, cô nàng muốn dãy ngoặc thu được là đúng. Tuy nhiên, cô tỏ ra khá bối rối vì có thể có nhiều cách điền khác nhau. Trong phút yếu lòng, cô muốn biết liệu có bao nhiêu cách điền, và cô sẵn sàng trả ơn cho ai tính được

Dữ liệu vào

Gồm một dòng duy nhất chứa một xâu S chỉ gồm các ký tự thuộc $\{', ', '[,]', '(', ')', '?'\}$ mô tả lại dãy ngoặc, các ký tự '?' mô tả các vị trí bị mờ

Kết quả

Một số nguyên duy nhất là số dư của số cách điền vào những chỗ bị mờ khi chia cho $10^9 + 7$. Hai cách điền được coi là khác nhau nếu dãy ngoặc thu được từ chúng khác nhau

Ví dụ

stdin	stdout
(?[??)	4
()[(())]{}	1
)????	0

Hạn chế

- 30% test có $|S| \leq 10$
- 70% test có $10 < |S| \leq 500$

Bài G. DGRAPH

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

Hệ thống mạng trên hành tinh XYZ thỏa mãn tính chất sau: Giữa hai đỉnh x, y bất kỳ, tồn tại và duy nhất một đường đi đơn giữa chúng và được ký hiệu là $\text{Path}(x, y)$. Nói cách khác, hệ thống mạng có dạng hình cây. Có một số cặp nút mạng đang truyền thông tin cho nhau, gọi là các kết nối. Với a là một kết nối, ký hiệu s_a và t_a lần lượt là đỉnh gửi và đỉnh nhận ($s_a \neq t_a$).

Do tính chất của mạng quang không lọc, thông tin được gửi đi từ một nút nào đó sẽ lan truyền khắp nơi. Các đỉnh nhận sẽ phân biệt gói tin dựa vào chiều truyền đến của gói tin và bước sóng của kết nối. Ta nói kết nối a làm nhiều kết nối b nếu tin từ s_a và từ s_b đến t_b theo cùng một chiều, cụ thể là $\text{Path}(s_a, t_b)$ và $\text{Path}(s_b, t_b)$ có cạnh chung.

Trên thực tế, việc gán bước sóng cho các kết nối sẽ đưa về bài toán tô màu trên đồ thị quan hệ "làm nhiều" nói trên. Tuy nhiên trong bài này, bạn chỉ cần tính số cung của đồ thị quan hệ đó, tức là số cặp a, b mà a làm nhiều b .

Các kết nối trên mạng có tính trực tuyến. Ban đầu chưa có kết nối nào, sau đó có thể có thêm các kết nối hoặc một số kết nối mất đi. Sau mỗi lần biến đổi như vậy, hãy tính toán và đưa ra số cặp a, b mà a làm nhiều b .

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương: n, m là số đỉnh của cây và số thay đổi của mạng
- n dòng tiếp theo mỗi dòng ghi một cạnh của cây: u, v
- m dòng tiếp theo mỗi dòng ghi một biến đổi của mạng: k, s, t với s là đỉnh gửi, t là đỉnh nhận, $k = 0/1$ tương ứng là có thêm hoặc mất đi một kết nối từ s đến t

Dữ liệu đảm bảo có ít nhất một kết nối từ s đến t khi $k = 1$, và nếu có nhiều kết nối từ s đến t thì mỗi lần chỉ mất đi một trong số đó. Các đỉnh của cây được đánh số từ 1.

Kết quả

- Ghi m dòng là kết quả tính được sau mỗi biến đổi

Ví dụ

stdin	stdout
7 5	0
1 2	1
1 3	4
2 4	8
2 5	3
3 6	
3 7	
0 2 7	
0 3 1	
0 1 4	
0 4 6	
1 2 7	

Hạn chế

- $1 \leq n, m \leq 10^5$
- Subtask 1: $1 \leq n, m \leq 1000$
- Subtask 2: $1 \leq m \leq 1000$

- Subtask 3: $1 \leq n \leq 1000$
- Subtask 4: Ràng buộc gốc

Bài H. TWOGRAPH

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

Kiến thường sống thành từng đé chế với tổ chức xã hội rất cao. Có hai đé chế kiến, đỏ và đen. Mỗi đé chế kiến được mô tả bởi một đồ thị vô hướng có trọng số. Do chỉ có một cáp viễn thông giữa hai đé chế, để truyền tải dữ liệu giữa thành phố của hai đé chế trước tiên dây cáp phải được di chuyển đến hai thành phố tương ứng. Giả sử hiện tại dây cáp đang nối thành phố u -đỏ và v -đen, ta nói dây cáp đang ở (u, v) . Việc chuyển cáp có thể gồm nhiều bước, mỗi bước thuộc một trong hai loại sau:

- Chuyển dây cáp từ (u, v) sang (x, v) . Việc này phải sử dụng một số tuyến đường của đé chế đỏ để đi từ u sang x . Tất cả các tuyến đường sử dụng đến đều phải đã được đóng phí bởi thành phố v
- Chuyển dây cáp từ (u, v) sang (u, y) . Việc này phải sử dụng một số tuyến đường của đé chế đen để đi từ v sang y . Tất cả các tuyến đường sử dụng đến đều phải đã được đóng phí bởi thành phố u

Mỗi thành phố của đé chế này cần phải chọn ra một số tuyến đường của đé chế kia để đóng phí, sao cho hệ thống đảm bảo mọi cặp (u, v) đều có thể được truyền tải dữ liệu với nhau. Lưu ý là khi một thành phố đóng phí sử dụng cho một tuyến đường nào đó, họ chỉ đóng phí một lần và được sử dụng tuyến đường đó nhiều lần, mức phí của mỗi tuyến đường là như nhau với mọi thành phố. Tuy nhiên việc thu phí của một tuyến đường có phân biệt các thành phố, tức là mỗi tuyến đường có thể sẽ được đóng phí bởi nhiều thành phố khác nhau. Hãy tính tổng phí nhỏ nhất có thể mà tất cả các thành phố phải đóng

Dữ liệu vào

Đồ thị mô tả hai đé chế sẽ được đọc vào lần lượt, mỗi đồ thị gồm có:

- Dòng đầu tiên chứa n m là số đỉnh và số cạnh
- m dòng tiếp theo mỗi dòng chứa u v w là thành phố hai đầu một tuyến đường và chi phí phải đóng để sử dụng tuyến đường đó

Kết quả

- Ghi ra tổng phí nhỏ nhất tìm được. Nếu không thể truyền tải dữ liệu cho mọi cặp, in ra -1

Ví dụ

stdin	stdout
6 9 1 4 52 2 1 100 1 3 94 6 4 9 6 3 97 5 6 17 2 6 35 6 3 15 2 3 37 8 7 3 2 90 1 6 68 5 8 79 7 3 71 1 3 47 7 5 15 6 4 30	1333

Hạn chế

- $1 \leq n, m \leq 10^5, 1 \leq w \leq 10^9$
- Có 50% số test với $1 \leq n, m \leq 1000$

07/11:

Bài I. BUYGOOD

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

Có m cửa hàng cùng bán một loại sản phẩm, cửa hàng thứ i có s_i sản phẩm giá mỗi sản phẩm là p_i và phải trả w_i tiền hóa đơn nếu mua sản phẩm của cửa hàng đó. Có nghĩa là nếu mua k sản phẩm của cửa hàng thứ i ($k \leq s_i$), bạn phải trả: 0 tiền nếu $k = 0$ hoặc $w_i + k \times p_i$ tiền nếu $k \neq 0$

Yêu cầu: Tính số tiền ít nhất để mua được n sản phẩm

Dữ liệu vào

- Dòng đầu tiên chứa n m
- m dòng tiếp theo mỗi dòng chứa 3 số s_i p_i w_i

Kết quả

- Số tiền ít nhất để mua được n sản phẩm

Ví dụ

stdin	stdout
20 4 5 5 6 10 4 12 15 6 9 20 7 0	118

Hạn chế

- $n \leq 10^4$, $m \leq 100$, $0 \leq s_i$, $p_i \leq 10^4$, $0 \leq w_i \leq 10^6$

Bài J. SSTORM

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

Đất nước xyz tươi đẹp được mô tả bằng một ma trận $n \times m$ ô vuông, ô ở dòng thứ i trên xuống và cột thứ j trái sang gọi là ô (i, j) . Khoảng cách giữa ô (x, y) với ô (i, j) là $|x - i| + |y - j|$. Ô (i, j) có giá trị tài sản là $a_{i,j}$

Do biến đổi khí hậu, những cơn bão ngày một nhiều hơn và mạnh hơn. Mỗi cơn bão được đặc trưng bởi 5 số nguyên không âm: w - cấp bão; R_1 - bán kính bão; R_2 - bán kính mắt bão; (x, y) - tọa độ bão sẽ đổ bộ. Theo đó, các ô (i, j) có khoảng cách đến ô (x, y) thuộc đoạn $[R_2; R_1]$ sẽ bị tác động, và giá trị tài sản ở ô đó sẽ giảm đi $\min(w, b_{i,j})$ với $b_{i,j}$ là giá trị tài sản hiện có ở ô (i, j)

Là một nước giáp biển, hằng năm đất nước xyz phải đón $k \leq 10^5$ trận bão. Do đặc trưng địa hình, bão sẽ chỉ đổ bộ vào một trong $q \leq 5$ điểm phân biệt. Rút kinh nghiệm từ siêu bão Hải Yến, ủy ban chống bão muốn biết sau khi k trận bão đổ tàn phá thì nước này phải chịu tổng thiệt hại là bao nhiêu? (Tổng thiệt hại của nước này được tính bằng tổng mức giảm giá trị tài sản của tất cả các ô)

Dữ liệu vào

- Dòng đầu tiên chứa 4 số nguyên: n m q k
- n dòng tiếp theo, mỗi dòng chứa m số. Số thứ j trên dòng i là $a_{i,j}$
- Dòng tiếp theo ghi $2q$ số nguyên là tọa độ của q điểm đặc biệt: x_1 y_1 x_2 y_2 \dots x_q y_q
- k dòng cuối, mỗi dòng ghi 5 số mô tả một cơn bão: w R_1 R_2 x y (Dữ liệu đảm bảo (x, y) là một trong q điểm đã cho)

Các trận bão được liệt kê theo đúng thứ tự sẽ đổ bộ

Kết quả

- Một số nguyên duy nhất: Tổng thiệt hại

Ví dụ

stdin	stdout
3 4 2 4 10 11 12 15 20 10 11 25 30 32 35 40 1 1 3 4 2 2 0 3 4 2 2 0 1 1 2 4 2 1 1 2 3 1 3 4	56

Hạn chế

- $1 \leq n, m \leq 500$, $0 \leq a_{i,j} \leq 10^9$, $0 \leq R_2 \leq R_1 \leq 1000$, $0 \leq w \leq 1000$
- 20% test có $q = 1$
- 60% test có $k \leq 10$

Bài K. FBIT

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2 giây

Những dãy nhị phân ản chứa một vẻ đẹp huyền bí (dãy nhị phân ở đây được hiểu là một xâu chỉ gồm các ký tự 0 và 1, các ký tự trên xâu được đánh số bắt đầu từ 1). Theo nghiên cứu mới đây, nếu một xâu nhị phân mà tìm thấy k ký tự 1 liên tiếp nhau thì đó là FBIT – một đại diện cho vẻ đẹp nữ thần tình ái. Tức là, xâu nhị phân S là FBIT nếu:

$$\exists i \in \{1, 2, \dots, |S| - k + 1\} | S_i = S_{i+1} = \dots = S_{i+k-1} = 1$$

Có một xâu nhị phân S độ dài n đã bị mờ đi một số chỗ. Chỉ còn lại m vị trí, được mô tả bởi m số nguyên khác 0. Mỗi số nguyên i có nghĩa:

- $i > 0$: $S_i = 0$
- $i < 0$: $S_{-i} = 1$

Nhà nghiên cứu muốn khôi phục lại xâu S bằng cách:

- Chọn một số nguyên $L \in \{1, 2, \dots, n\}$, cắt bỏ phần cuối S chỉ giữ lại L vị trí đầu tiên
- Điền các ký tự 0 hoặc 1 vào những chỗ bị mờ đi của xâu còn lại để thu được FBIT

Hãy giúp ông ta tính toán số cách khôi phục. Hai cách khôi phục được coi là khác nhau nếu FBIT thu được từ chúng khác nhau. Do kết quả có thể sẽ rất lớn, chỉ cần in ra phần dư khi chia cho $10^9 + 7$

Dữ liệu vào

- Dòng đầu tiên chứa n m k
- Dòng thứ 2 chứa m số nguyên mô tả các vị trí chưa bị mờ của xâu S . Dữ liệu đảm bảo không có 2 số nào có giá trị tuyệt đối bằng nhau

Kết quả

- Một số nguyên duy nhất – kết quả bài toán

Ví dụ

stdin	stdout
10 3 2 3 -6 8	235

Hạn chế

- 50% test với $1 \leq k \leq n \leq 1000$
- 50% test với $1000 < n \leq 10^{18}$. $1 \leq k \leq 10$. $1 \leq m \leq 1000$

Bài L. VIRUS

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

Hùng đang phát triển một phần mềm diệt virus dựa trên nguyên tắc so khớp mẫu. Theo đó, cậu có tập S các xâu nhị phân là mẫu thường gặp trong mã nguồn của virus (mã độc). Mỗi tệp tin f có thể được hiểu như một xâu nhị phân, độ tương thích của f với mẫu x là số lần xuất hiện của x ở trong f (xuất hiện theo nghĩa bằng với một đoạn con liên tiếp). Để đánh giá f có bị nhiễm virus hay không, Hùng muốn tính tổng độ tương thích của f với mọi xâu trong S .

Trong quá trình sử dụng, Hùng có thể cập nhật thêm các mẫu virus mới. Ban đầu Hùng chưa có mẫu virus nào được báo cáo. Cụ thể bạn cần thực hiện hai loại truy vấn sau:

- Loại một: 0 x : Thêm xâu x vào tập S . Nếu x đã xuất hiện trong S , Hùng vẫn thêm x vào (theo cậu, càng có nhiều người dùng báo cáo một mẫu thì mẫu đó càng quan trọng). Có thể hiểu S là một multiset.
- Loại hai: 1 f : Tính toán và đưa ra tổng số lần xuất hiện của các xâu trong S trên xâu f .

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương Q
- Q dòng tiếp theo mỗi dòng mô tả một truy vấn: 0 x hoặc 1 f

Kết quả

- Với mỗi truy vấn loại hai, in ra kết quả trên một dòng

Ví dụ

stdin	stdout
5	10
0 0	11
0 1	
0 01	
1 10011001	
1 01010011	

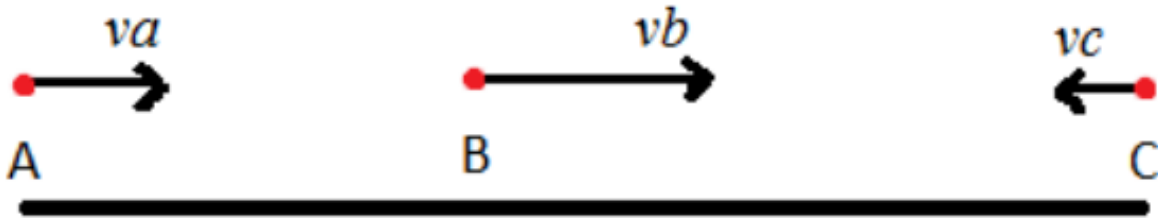
Hạn chế

- $1 \leq Q \leq 10^6$, tổng độ dài tất cả các xâu đầu vào không quá 2×10^6
- Subtask 0: $|S| \leq 10$
- Subtask 1: Các truy vấn loại hai nằm liên tiếp nhau
- Subtask 2: Ràng buộc gốc

Bài M. FLYBIRD

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

Trên một đường thẳng có 2 con chim A và C đang bay về phía nhau. Một con ruồi ở B bay về phía C. Vận tốc của các con vật được cho trước và vận tốc của ruồi là lớn nhất. Được biết rằng ruồi rất sợ chim, nên khi B và C gặp nhau tại điểm D nào đó, B sẽ đổi hướng và bay về phía A. Sau khi B gặp A tại E nào đó, B sẽ lại đổi hướng, cứ thế lặp lại mãi. Hãy tính tổng độ dài các quãng đường mà B đã bay, sai số không quá 10^{-4}



Dữ liệu vào

- Gồm 5 số nguyên dương không quá 10^9 : độ dài AB, độ dài BC, vận tốc va, vb, vc

Kết quả

- Một số thực ghi kết quả, quy tròn đến 4 chữ số thập phân

Ví dụ

stdin	stdout
8 5 5 6 4	8.6667

Bài N. MMONEY

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

Trong cây ATM có n loại tiền, loại thứ i có số lượng a_i (tờ) và giá trị b_i (đồng). Một người đến cây và muốn rút ra S đồng

Hãy giúp máy đếm xem có bao nhiêu cách trả tiền thỏa mãn. Hai cách được coi là khác nhau nếu tồn tại một loại tiền được rút với số lượng tờ khác nhau trong hai cách đó. Lưu ý là b_i không nhất thiết phân biệt, tuy nhiên ta vẫn xem n loại tiền đó là khác nhau

Dữ liệu vào

- Dòng đầu: n S
- n dòng tiếp theo, dòng thứ i ghi: a_i b_i

Kết quả

- Một số nguyên là kết quả bài toán, sau khi chia lấy dư cho 1000000007

Ví dụ

stdin	stdout
3 10 3 1 3 2 3 3	5

Hạn chế

- $1 \leq n, S, a_i, b_i \leq 6000$
- Subtask 1: $n, S \leq 200$
- Subtask 2: $n, S \leq 2000$
- Subtask 3: Ràng buộc gốc
- Có 50% số test có b_i phân biệt

Bài O. GCLEAR

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

Năm 2060, công nghệ của con người đã đạt được những thành tựu vượt bậc. Robot giờ đây đã rất phổ biến, chúng giúp đỡ con người trong hầu hết các công việc hàng ngày. Các robot dọn vệ sinh được bày bán khắp nơi

Những robot này được lập trình sẵn để hút bụi trong các căn phòng. Để robot hiểu được, căn phòng sẽ được chia thành lưới ô vuông $n \times m$, các dòng đánh số từ 1 đến n từ trên xuống dưới, các cột đánh số từ 1 đến m từ trái sang phải. Các vật dụng trong nhà sẽ được xem là các vật cản trên lưới, robot không thể đi vào. Để di chuyển trong căn phòng, chúng ta có bốn lệnh điều khiển, là sang trái, lên trên, sang phải, hay xuống dưới. Tức là robot sẽ đi sang ô kề cạnh với ô đang đứng (trên lưới, không có vật cản). Để hút bụi, có một lệnh là hút bụi, robot sẽ thực hiện hút bụi ở ngay ô đang đứng

Hiện tại có một robot đã được lập trình sẵn để hút bụi trong một căn phòng. Tuy nhiên chủ nhân của nó lại đổi ý, ông không muốn nó làm sạch căn phòng vốn quen thuộc của mình nữa. Giải pháp của ông là mua một robot khác, gọi là “robot cản phá”, robot này không có khả năng hút bụi, nó chỉ được lập trình để di chuyển trong căn phòng. Quy cách lập trình để di chuyển của nó cũng giống như robot kia. Sau đó đồng thời khởi động 2 robot. Trước khi khởi động robot hút bụi, ông có một ngày để lập trình cho robot cản phá, để nó sẽ nhanh chóng bắt được robot hút bụi nhất (thời điểm mà hai robot nằm vào cùng một ô là sớm nhất). Vì giống nhau về quy cách lập trình, chúng sẽ làm việc đồng thời, tức là robot này thực hiện một lệnh thì robot kia cũng thực hiện một lệnh, cùng thời điểm, mất một đơn vị thời gian

Cho cách lập trình của robot hút bụi, liệu bạn có lập trình được robot cản phá?

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên: n m
- Tiếp theo là một ma trận n dòng m cột, mô tả về các vật cản trong phòng. Ký tự thứ j trên dòng i là 0/1 tương ứng là ô i j không có/có vật cản
- Dòng tiếp theo ghi 2 số nguyên x y là vị trí sẽ đặt robot hút bụi
- Tiếp theo là một chuỗi S mô tả lập trình cho robot hút bụi, gồm các ký tự "L", "R", "U", "D", "C" tương ứng là sang trái, sang phải, lên trên, xuống dưới, hút bụi
- Dòng cuối cùng ghi 2 số nguyên z t là vị trí sẽ đặt robot cản phá

Kết quả

- Nếu không thể cản phá được (tức dù ông lập trình robot cản phá ra sao, thì robot hút bụi cũng có thể thực hiện hết chuỗi lệnh của mình), in ra -1. Ngược lại, in ra thời điểm sớm nhất cản phá được

Ví dụ

stdin	stdout
4 4 0000 0000 0000 0000 4 4 UUCLDCL 1 1	6
3 2 00 00 00 3 2 UDUDUDUDUDUDUDUDUDUDUDLRLRLRUUDCC 1 1	34
4 5 -1 00100 00100 01000 00000 4 5 UUCLDCL 1 1	-1

Hạn chế

- $1 \leq n, m \leq 100, 1 \leq |S| \leq 1000$
- Có 50% test ứng với $1 \leq n, m \leq 50$

Bài P. FPAIR

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

Một câu lạc bộ lập trình có nhiều thành viên rải rác khắp thành phố. Sắp tới họ muốn ghép đôi với nhau để làm việc hiệu quả hơn. Bản đồ thành phố là một đơn đồ thị vô hướng liên thông, gồm n địa điểm và m con đường hai chiều nối các địa điểm này với nhau. Có k thành viên trong câu lạc bộ đánh số từ 1 đến k , người thứ i sống tại địa điểm p_i (không nhất thiết phân biệt)

Vì ngại di chuyển, mỗi người trong số họ đều muốn ghép đôi với người gần họ nhất, nếu có nhiều người gần nhất thì người đó sẽ chọn người có chỉ số nhỏ nhất. Cho biết bản đồ thành phố. Với mỗi thành viên, hãy xác định xem anh ta muốn ghép đôi với người nào

Dữ liệu vào

- Dòng đầu tiên chứa n m k
- Dòng tiếp theo chứa dãy p
- m dòng tiếp theo, mỗi dòng chứa một cạnh: u v w

Kết quả

- Gồm k số nguyên là kết quả cho k người

Ví dụ

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

Hạn chế

- $1 \leq n, m, k \leq 10^5$, $0 \leq w \leq 10^4$
- Có 30% test với $n, m, k \leq 1000$

08/11:

Bài Q. ICNT

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

Cho dãy số nguyên a gồm n số nguyên. Số nghịch thế của a là số cặp i, j sao cho $1 \leq i < j \leq n$ và $a_i > a_j$. Ta thực hiện Q biến đổi trên a , mỗi biến đổi có dạng:

Tăng a_i lên một lượng bằng x .

Yêu cầu in ra sau số nghịch thế của a sau mỗi truy vấn

Dữ liệu vào

- Dòng đầu chứa n và Q
- Dòng tiếp theo chứa n số tự nhiên: a_1, a_2, \dots, a_n
- Q dòng tiếp theo mỗi dòng chứa i và s . Công thức tính x là: $x = |s + \text{sum}| \cdot (2 \times 10^9 + 1) - 10^9$. Ở đây sum là tổng các kết quả của các truy vấn trước truy vấn này

Kết quả

Gồm Q dòng ghi kết quả tương ứng

Ví dụ

stdin	stdout
4	1
1 2 3 4	1
2	
3 999999998	
3 1000000003	

Hạn chế

- $1 \leq n, Q \leq 10^5$, $0 \leq s \leq 2 \times 10^9$, $-10^9 \leq a_i \leq 10^9$ trong mọi thời điểm
- 30% test: $n, Q \leq 10^4$

Bài R. XAMBQ

File dữ liệu vào: XAMBQ.inp
File kết quả: XAMBQ.out
Hạn chế thời gian: 1 giây

Cho ma trận A có kích thước $n \times n$ chứa các giá trị trong khoảng 0 đến 1000. Tiến hành xây dựng ma trận B như sau:

Với (i, j) là một ô trên ma trận, xét mọi đường đi từ $A_{1,1}$ đến $A_{i,j}$ chỉ được đi sang phải và xuống dưới. Tiến hành tính tổng các số đi qua trên mỗi đường đi và chọn ra đường có tổng lớn nhất, gọi tổng đó là $B_{i,j}$. Một cách dễ hiểu hơn, $B_{i,j}$ là tổng trọng số lớn nhất có thể khi đi từ ô $(1, 1)$ đến ô (i, j) mà chỉ được đi sang phải (cột tăng một) hoặc xuống dưới (dòng tăng một).

Bài toán có vẻ đơn giản, nhưng A là một ma trận cà khịa. Thật vậy, A có thể thực hiện các biến đổi dạng UP($A_{x,y}$) hoặc DOWN($A_{x,y}$) tương ứng là tăng hoặc giảm $A_{x,y}$ đúng một đơn vị. Bạn cần tính lại ma trận B sau mỗi lần cà khịa như vậy. Tất nhiên là có quá nhiều số để in ra, vì vậy bạn chỉ cần in ra tổng của ma trận B sau khi tính lại mà thôi

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên: n
- Tiếp theo là n dòng mỗi dòng n số nguyên mô tả A : $A_{x,y}$
- n dòng cuối ghi n thay đổi của A : $c \ x \ y$ với c bằng U hoặc D

Kết quả

- Gồm $n + 1$ dòng, dòng đầu tiên ghi tổng của ma trận B trước khi thay đổi ma trận A , mỗi dòng trong n dòng tiếp theo ghi tổng của ma trận B sau mỗi phép thay đổi của A

Ví dụ

XAMBQ.inp	XAMBQ.out
5	420
3 3 5 0 5	445
4 4 2 4 3	453
4 4 2 1 3	451
4 3 2 4 2	448
2 3 3 2 4	442
U 1 1	
U 1 3	
D 5 2	
D 4 3	
D 2 4	

Hạn chế

- $1 \leq n \leq 1500$
- Có 50% số test với $n \leq 500$
- Độ lớn của n sẽ được rải đều cho các test

Bài S. DOPA

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

Một xâu được gọi là đối xứng kép nếu nó có thể tách làm hai xâu đối xứng (một trong hai xâu đó có thể rỗng). Nói cách khác, xâu đối xứng kép là xâu đối xứng hoặc được ghép từ hai xâu đối xứng. Ví dụ xâu banana, abcba là các xâu đối xứng kép. Hãy đếm số xâu đối xứng kép độ dài không quá n , chỉ chứa các ký tự latin thường

Dữ liệu vào

- Dòng duy nhất chứa số nguyên dương n

Kết quả

- Ghi kết quả sau khi chia lấy dư cho $10^9 + 7$

Ví dụ

stdin	stdout
3	2678
4	38454

Hạn chế

- $1 \leq n \leq 10^5$
- 20% test có $n \leq 15$
- 20% test có $n \leq 1000$

Bài T. BEULER

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

Cho đồ thị vô hướng G , mỗi cạnh có hai trọng số là chi phí đi lại theo 2 chiều. Cần tìm một đường đi xuất phát từ 1, đi qua tất cả các cạnh, mỗi cạnh đúng một lần và quay về 1, sao cho chi phí lớn nhất trong số các chi phí trên các chiều của các cạnh đã đi qua là nhỏ nhất có thể.

Dữ liệu vào

- Dòng đầu ghi số đỉnh và số cạnh của đồ thị: n m
- m dòng tiếp theo, mỗi dòng ghi một tuyến đường: u v c_{uv} c_{vu}

Kết quả

Nếu không tồn tại cách đi, in ra -1. Ngược lại in ra trên 2 dòng:

- Dòng đầu ghi trọng số lớn nhất trên đường đi tìm được.
- Dòng tiếp theo ghi m số là chỉ số của các cạnh sẽ đi qua theo thứ tự. Cách cạnh đánh số từ 1 theo thứ tự đọc vào

Ví dụ

stdin	stdout
4 6 1 2 2 4 2 3 3 4 3 4 4 4 4 1 5 4 1 3 1 10 1 3 1 10	5 5 2 1 6 3 4

Hạn chế

- $1 \leq n, m \leq 1000$

Bài V. RECOLOR

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

Một cách tô màu trên đồ thị là một cách gán cho mỗi đỉnh một màu sao cho các đỉnh kề nhau thì khác màu. Cho đồ thị vô hướng G và một cách tô màu G bằng k màu. Hãy tìm cách tô màu lại G (có thể giống hoặc khác cách tô đã cho) bằng x màu ($1 \leq x \leq k$) sao cho tồn tại một đường đi đơn độ dài x chứa tất cả các màu.

Dữ liệu vào

- Dòng đầu chứa số đỉnh, số cạnh, số màu: $n \ m \ k$ ($n, m, k \leq 10^5$)
- Dòng thứ hai chứa danh sách màu: $c_1 \ c_2 \ \dots \ c_n$
- Mỗi dòng trong số m dòng tiếp theo chứa một cạnh: $u \ v$

Các đỉnh của đồ thị được đánh số từ 1 đến n , các màu được dùng là $\{1, 2, 3, \dots, k\}$

Kết quả

- Dòng đầu ghi số màu và danh sách màu: $x \ c'_1 \ c'_2 \ \dots \ c'_n$
- Dòng tiếp theo ghi x số là các đỉnh theo thứ tự trên đường đi tìm được

Lưu ý là bạn không cần cực tiểu x .

Ví dụ

stdin	stdout
3 3 3 1 2 3 1 2 2 3 3 1	3 3 2 1 1 2 3

Bài W. INCPATH

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

Cho đồ thị có hướng n đỉnh m cạnh. Các đỉnh được đánh số v_1, v_2, \dots, v_n . Cần gán cho mỗi đỉnh một giá trị là một số nguyên từ 1 đến n (hai đỉnh khác nhau được gán hai giá trị khác nhau) sao cho mọi đường đi đơn xuất phát từ v_1 đều đi qua các đỉnh theo thứ tự tăng dần của giá trị

Dữ liệu vào

- Dòng đầu chứa: n m
- m dòng tiếp theo, mỗi dòng chứa: i j cho biết đỉnh v_i được nối với v_j bằng một cung

Kết quả

- In -1 nếu không tồn tại cách gán, ngược lại in ra n số là giá trị của v_1, v_2, \dots, v_n

Ví dụ

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

Hạn chế

- $1 \leq n, m \leq 1000$
- Có ít nhất 50% số test với $n \leq 10$

Bài X. LTC

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

Cho một cây n đỉnh và m đường đi trên cây. Hãy đếm xem có bao nhiêu cách chọn ra một số đường trong số m đường đi đó sao cho giao của các đường đi này khác rỗng. Hai cách chọn được coi là khác nhau nếu tồn tại một đường được chọn trong cách này nhưng không được chọn trong cách kia. Lưu ý là không chọn đường nào cũng là một cách chọn hợp lệ

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương: n m
- $n - 1$ dòng tiếp theo mỗi dòng chứa một cạnh của cây: u v
- m dòng tiếp theo mỗi dòng chứa một đường đi trong tập đã cho, gồm chỉ số của hai đỉnh mút của đường đi đó: x y

Kết quả

- Chứa số cách chọn tính được, chỉ cần in ra phần dư khi chia cho $10^9 + 7$

Ví dụ

stdin	stdout
3 3 1 2 1 3 1 2 1 3 2 3	8

Hạn chế

- $1 \leq n, m \leq 10^5$
- Subtask 1: $n, m \leq 1000$
- Subtask 2: Ràng buộc gốc