

Chia kẹo

Alice có n chiếc kẹo và muốn chia n chiếc kẹo vào các túi để trao thưởng cho các bạn nhỏ, mỗi túi sẽ tương ứng với một mức thưởng. Cụ thể, với hai số nguyên dương n, k , Alice muốn chia kẹo thỏa mãn:

- ✓ Số lượng túi ít nhất là 2 túi;
- ✓ Tổng số kẹo trong các túi bằng đúng n ;
- ✓ Chênh lệch số kẹo giữa hai túi bất kì lớn hơn k .

Ví dụ, với $n = 9, k = 1$, Alice có 4 cách chia như sau: $1 + 8; 2 + 7; 3 + 6; 1 + 3 + 5$.

Yêu cầu: Cho ba số nguyên dương n, k, M . Gọi C là số cách chia kẹo thỏa mãn, hãy tính $C \% M$, trong đó $\%$ là phép toán chia lấy dư.

Dữ liệu: Gồm một dòng chứa ba số nguyên dương n, k, M ($k \leq n \leq 10^5; M \leq 10^9 + 9$).

Kết quả: Gồm một số nguyên là phần dư trong phép chia C cho M .

Ràng buộc:

- Có 30% số test ứng với 30% số điểm của bài có $n \leq 20$;
- Có 30% số test khác ứng với 30% số điểm của bài có $n \leq 200$;
- Có 20% số test khác ứng với 20% số điểm của bài có $n \leq 2000$;
- Có 20% số test còn lại ứng với 20% số điểm của bài có $n \leq 10^5$.

Ví dụ:

Input	Output
9 1 100	4

Liên thông

Một khu du lịch do Alice quản lí là một quần đảo được mô tả bằng một bảng kí tự gồm m hàng và n cột, các hàng được đánh số từ 1 đến m từ trên xuống dưới, các cột được đánh số từ 1 đến n từ trái sang phải. Ô nằm giao giữa hàng i và cột j gọi là ô (i, j) và chứa kí tự '1' hoặc '0', trong đó '1' mô tả vùng đất (gọi là ô đất), '0' mô tả vùng nước (gọi là ô nước). Một đảo là một vùng liên thông gồm các ô đất mà hai ô có thể đi được đến nhau thông qua các ô đất chung cạnh. Khách du lịch có thể chọn xuất phát tại một ô đất và đi bộ thăm hết các ô đất trên đảo (chứa ô đất xuất phát) bằng cách di chuyển thông qua các ô chung cạnh.

Với mong muốn khách du lịch có thể đi bộ và thăm quan được nhiều hơn, Alice lên kế hoạch dựng cầu treo trên các ô nước để liên thông một số đảo. Cụ thể, sau khi dựng cầu treo ở các ô nước, du khách có thể di chuyển vào các ô nước đó như đi vào các ô đất, Alice mong muốn tồn tại cách thăm quan cho du khách mà tổng số ô đất du khách có thể đi thăm quan được lớn hơn hoặc bằng s và số lượng ô nước cần dựng cầu treo là ít nhất.

Yêu cầu: Cho bảng kí tự mô tả quần đảo và T phương án tương ứng với T số nguyên dương s_1, s_2, \dots, s_T , với mỗi phương án s_k ($1 \leq k \leq T$) hãy giúp Alice tính số ô nước ít nhất cần dựng cầu treo để tổng số ô đất mà khách du lịch có thể đi bộ và thăm được lớn hơn hoặc bằng s_k .

Dữ liệu:

- Dòng đầu gồm ba số nguyên dương m, n, T ($T \leq 100$);
- Dòng thứ i ($1 \leq i \leq m$) trong số m dòng tiếp theo chứa một xâu độ dài n chỉ gồm hai loại kí tự '1' và '0';
- Dòng cuối cùng chứa T số nguyên dương s_1, s_2, \dots, s_T ($s_k \leq m \times n$).

Kết quả:

- Gồm T số nguyên tương ứng với T phương án, mỗi số là số ô nước ít nhất cần dựng cầu treo hoặc đưa ra -1 nếu không tồn tại cách dựng thỏa mãn.

Ràng buộc:

Gọi c là số đảo trong khu du lịch, trong tất cả các test đều thỏa mãn $1 \leq c \leq 10$ và $m, n \leq 100$, ngoài ra:

- Có 10% số test ứng với 10% số điểm của bài có $c = 1; m, n \leq 30$;
- Có 25% số test ứng khác với 25% số điểm của bài có $c \leq 2; m, n \leq 30$;
- Có 25% số test ứng khác với 25% số điểm của bài có $c \leq 3; m, n \leq 30$;
- Có 20% số test ứng khác với 20% số điểm của bài $c \leq 8; m, n \leq 30$;
- Có 20% số test còn lại ứng với 20% số điểm của bài không có ràng buộc nào thêm.

Ví dụ:

Input	Output	Input	Output
5 6 3 000001 011101 010011 000001 001100 9 10 15	1 2 -1	3 5 6 10101 01010 10101 1 2 3 4 5 6	0 1 1 1 2 2

Chiếu sáng

Quảng trường trong khu du lịch mà Alice quản lý là một sân rộng hình vuông với tọa độ góc trái dưới là $(0,0)$ và góc phải trên là $(10^6, 10^6)$. Có n máy chiếu sáng laser, máy thứ i ($1 \leq i \leq n$) khi chiếu sẽ xuất hiện một vệt sáng là một đoạn thẳng có tọa độ đầu mút là (x_i, y_i) và (u_i, v_i) trên quảng trường. Biết rằng, không có hai máy nào khi cùng chiếu sáng tạo ra hai vệt sáng là hai đoạn thẳng thỏa mãn: hai đoạn thẳng cùng nằm trên một đường thẳng và có điểm chung. Alice muốn tạo ra các kịch bản chiếu sáng, cô cần đếm số cách chọn 4 máy để khi cả 4 máy cùng chiếu sáng thì sẽ xuất hiện một hình chữ nhật tạo bởi các vệt sáng với 4 góc là 4 giao điểm của 4 đoạn thẳng.

Yêu cầu: Cho thông tin về n đoạn thẳng mô tả vệt sáng của các máy chiếu laser, hãy giúp Alice đếm số cách chọn thỏa mãn.

Dữ liệu:

- Dòng đầu chứa số nguyên dương n ($n \leq 5000$);
- Dòng thứ i ($1 \leq i \leq n$) trong n dòng sau chứa bốn số x_i, y_i, u_i, v_i ($0 \leq x_i, y_i, u_i, v_i \leq 10^6$).

Kết quả: Gồm một dòng chứa một số nguyên không âm là số cách chọn thỏa mãn.

Ràng buộc:

- Có 20% số test ứng với 20% số điểm của bài thỏa mãn: $n \leq 50$ và các đoạn thẳng đều song song hoặc vuông góc với cạnh quảng trường;
- Có 15% số test khác ứng với 15% số điểm của bài thỏa mãn: $n \leq 50$;
- Có 20% số test khác ứng với 20% số điểm của bài thỏa mãn: $n \leq 500$ và các đoạn thẳng đều song song hoặc vuông góc với cạnh quảng trường;
- Có 10% số test khác ứng với 10% số điểm của bài thỏa mãn: $n \leq 500$;
- Có 20% số test khác ứng với 20% số điểm của bài thỏa mãn: Các đoạn thẳng đều song song hoặc vuông góc với cạnh quảng trường;
- Có 15% số test còn lại ứng với 15% số điểm của bài mãn: Không có ràng buộc nào thêm.

Ví dụ:

Input	Output	Minh họa
10 30 20 30 80 10 20 100 20 10 40 100 40 10 60 30 60 10 20 10 80 50 60 100 60 30 40 50 60 30 40 50 20 90 20 50 60 90 60 50 20	4	

Truy vấn đoạn

Có n đoạn số nguyên trong đoạn $[1, M]$ nhưng bạn không biết chính xác các đoạn. Bạn cần tìm số đoạn độc lập rời nhau nhiều nhất bằng cách sử dụng không quá q truy vấn `query(L, R)`, hàm này trả về số đoạn nằm trong đoạn $[L, R]$.

Cụ thể, bạn cần xây dựng hàm `int solve(n, q)`, hàm trả về số đoạn độc lập rời nhau nhiều nhất, hàm này có thể gọi hàm dưới đây:

- `int query(L, R)`, hàm trả số đoạn $[a, b]$ trong n đoạn mà $L \leq a \leq b \leq R$.

Chú ý, hàm này không adaptive.

Subtask 1: $n \leq 1000$; $M \leq 10^3$; $q = 10^6$

Subtask 2: $n \leq 25000$; $M \leq 10^5$; $q = 10^5$

Subtask 3: $n \leq 25000$; $M \leq 10^9$; $q = 7.5 \times 10^5$

Subtask 4: $n \leq 25000$; $M \leq 10^9$; $q = 4.4 \times 10^5$

File `lrq.h`

```
// Cần cài đặt hàm
int solve(int n, int q);

// Sử dụng hàm
int query(int L, int R);
```