

Bài A. P2STR

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

Cho một chuỗi s chỉ chứa các ký tự latin thường. Đếm số bộ (i, j, k, t) thỏa mãn $1 \leq i \leq j < k \leq t \leq |S|$ và $s_i s_{i+1} \dots s_j s_k s_{k+1} \dots s_t$ là một chuỗi đối xứng.

Dữ liệu vào

Ghi một chuỗi s .

Kết quả

Ghi một số nguyên là kết quả bài toán.

Ví dụ

<code>stdin</code>	<code>stdout</code>
abbaca	14

Hạn chế

- Có 12% số test với $1 \leq |S| \leq 50$;
- Có 28% số test với $1 \leq |S| \leq 500$;
- Có 60% số test với $1 \leq |S| \leq 5000$;

Bài B. P3TREE

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

Cho một cây n đỉnh, các đỉnh được đánh số từ 1 đến n . Một bộ k đỉnh (x_1, x_2, \dots, x_k) được gọi là đẳng cấu thứ tự với hoán vị (p_1, p_2, \dots, p_k) nếu tồn tại một đường đi đơn đi qua các đỉnh x_1, x_2, \dots, x_k theo đúng thứ tự đó, và với mọi $1 \leq i < j \leq k$ thì $x_i < x_j$ khi và chỉ khi $p_i < p_j$. Cho $p = (p_1, p_2, p_3)$ là một hoán vị của $(1, 2, 3)$, hãy đếm số bộ 3 đỉnh đẳng cấu thứ tự với p .

Dữ liệu vào

- Dòng đầu ghi số nút của cây: n ;
- Dòng tiếp theo ghi p_1, p_2, p_3 ;
- $n - 1$ dòng tiếp theo, mỗi dòng ghi u, v là một cạnh của cây.

Kết quả

Ghi một số nguyên duy nhất là số bộ đẳng cấu thứ tự với p .

Ví dụ

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

Hạn chế

- Trong tất cả các test: $1 \leq n \leq 10^5$;
- Có 8% số test với $n \leq 500$;
- Có 12% số test với $n \leq 5000$;
- Có 28% số test với mỗi đỉnh đều kề với nhiều nhất 2 đỉnh khác;
- Có 52% số test với ràng buộc gốc.

Bài C. FXOR2

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

Cho dãy số nguyên $a = a_1, a_2, \dots, a_n$ và q truy vấn. Mỗi truy vấn có dạng (L, R, x) , cần tìm i sao cho $L \leq i \leq R$ và $a_i \wedge x$ đạt giá trị lớn nhất. Các số a_i và x đều được cho dưới dạng nhị phân.

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương n q ;
- Dòng thứ i trong n dòng tiếp theo chứa a_i ở dạng nhị phân;
- q dòng tiếp theo, mỗi dòng chứa L R x trong đó x ở dạng nhị phân;

Tất cả các số ngoài a_i và x đều được cho ở dạng thập phân.

Kết quả

Với mỗi truy vấn, in ra kết quả trên một dòng. Nếu có nhiều i thỏa mãn $a_i \wedge x$ đạt giá trị lớn nhất thì in ra i nhỏ nhất có thể.

Ví dụ

stdin	stdout
5 4	2
100	5
101	3
1	5
1011	
11	
2 3 10	
1 5 1100	
3 5 1010	
1 5 11100	

Hạn chế

- Trong tất cả các test: $n, q \leq 10^5$; tổng độ dài tất cả các xâu a_i không quá 10^6 ; tổng độ dài tất cả các xâu x không quá 10^6 ;
- Có 8% số test với $n, q \leq 5000$; độ dài các xâu nhị phân đều không quá 30;
- Có 12% số test với $n, q \leq 5000$;
- Có 28% số test với độ dài các xâu nhị phân đều không quá 30;
- Có 52% số test với ràng buộc gốc.

Bài D. STTREE

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

Cho tập S các điểm trên mặt phẳng ($2 \leq |S| \leq 100$). Các điểm được đánh số từ 0 đến $|S| - 1$, điểm thứ i có tọa độ (x_i, y_i) là hai số thực có giá trị tuyệt đối không quá 1000. Có hai thao tác có thể được thực hiện: Chèn thêm một điểm có tọa độ tùy ý vào S , nối hai điểm thuộc S với nhau bằng một đoạn thẳng.

Yêu cầu: Thực hiện không quá 10^5 thao tác như trên sao cho khi kết thúc quá trình, các điểm trong S liên thông và tổng độ dài các đoạn thẳng được nối là càng nhỏ càng tốt.

Dữ liệu vào

- Dòng đầu ghi $|S|$;
- $|S|$ dòng tiếp theo, dòng thứ i ghi: $x_i \ y_i$.

Kết quả

Dễ thấy các thao tác loại 1 có thể chuyển lên đầu. Tiến hành đánh số cho các điểm theo thứ tự được thêm, bắt đầu từ $|S|$. Định dạng đầu ra như sau:

- Dòng đầu ghi tổng độ dài các đoạn thẳng đã nối (ít nhất 3 chữ số thập phân);
- Dòng tiếp theo ghi k là số thao tác loại 1;
- Dòng thứ i trong k dòng tiếp theo ghi tọa độ của điểm thứ $|S| + i$;
- Dòng tiếp theo ghi q là số thao tác loại 2;
- Mỗi dòng trong q dòng tiếp theo ghi hai số $i \ j$ là chỉ số của hai điểm được nối.

Ví dụ

stdin	stdout
4	7.65685425
0 0	2
0 2	1 1
4 2	3 1
4 0	5
	0 4
	1 4
	4 5
	5 2
	5 3

Hạn chế

Gọi J là kết quả của giám khảo, P là kết quả của thí sinh. Điểm của thí sinh sẽ là $\min(1, \max(0, \frac{2J-P}{J}))$.