

Mục lục

Xếp lịch thí nghiệm — MACHINE	1
Liệt kê hoán vị - PERMUTATIONLIST	2
Thành phần liên thông mạnh - SCC	3
Bao đa giác - POLPACK	4
Liệt kê xâu nhị phân - BINARYLIST	5
Bản vanxơ Fibonacci — FIBVAL	6
Chuỗi từ Fibonacci — FIBWORDS	7
Thêm cạnh - ADDEDGE	8

Bài A. Xếp lịch thí nghiệm

File dữ liệu vào: `machine.inp`
File kết quả: `machine.out`
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 256 MB

Một kỹ sư cần chọn ra 2 giai đoạn trong số N giai đoạn cho trước để làm thí nghiệm sản xuất ra chất C . Mỗi giai đoạn i , $1 \leq i \leq n$ được cho bởi thời điểm bắt đầu s_i và thời điểm kết thúc t_i ($s_i < t_i$). Vì lý do kỹ thuật nên hai giai đoạn được chọn không được phép giao nhau, (hai giai đoạn i và j là không giao nhau nếu $t_i < s_j$ hoặc $t_j < s_i$). Nếu thí nghiệm chạy vào giai đoạn i thì lượng chất C được sản xuất ra sẽ bằng $t_i - s_i$ đơn vị.

Yêu cầu: Hãy giúp anh kỹ sư chọn được hai giai đoạn không giao nhau sao cho tổng lượng chất C sản xuất được là lớn nhất..

Dữ liệu vào

- Dòng 1: chứa một số nguyên n ($2 \leq n \leq 10^6$)
- Dòng $i + 1$: chứa hai số nguyên dương s_i và t_i ($s_i < t_i \leq 3 \times 10^6$)

Kết quả

Ghi ra duy nhất một số nguyên là lượng chất C thu được. Ghi ra -1 nếu không chọn được hai giai đoạn nào không giao nhau.

Ví dụ

<code>machine.inp</code>	<code>machine.out</code>
5 8 12 6 11 3 9 2 5 1 4	8

Giải thích

Hai giai đoạn lớn nhất không giao nhau là $[2, 5]$ và $[6, 11]$ sẽ thu được 8 đơn vị chất C .

Bài B. Liệt kê hoán vị

File dữ liệu vào: PERMUTATIONLIST.inp
File kết quả: PERMUTATIONLIST.out
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 256 MB

Cho 1 một số n . Đưa ra hoán vị độ dài n thứ k trong thứ tự từ điển.

Dữ liệu vào

Dòng đầu ghi 2 số nguyên dương n, k .

Kết quả

Ghi ra hoán vị thứ k trên một dòng duy nhất, các thành phần cách nhau bởi dấu cách. Nếu không tồn tại thì ghi ra -1.

Ví dụ

PERMUTATIONLIST.inp	PERMUTATIONLIST.out
1 2	-1
3 4	2 31

Hạn chế

$$n \leq 10^4$$

- Subtask 1 : $k \leq 10^4$
- Subtask 2 : $k \leq 10^9$

Bài C. Thành phần liên thông mạnh

File dữ liệu vào: `scc.inp`
File kết quả: `scc.out`
Hạn chế thời gian: 1 giây
Hạn chế bộ nhớ: 256 MB

Bài toán yêu cầu tính số thành phần liên thông mạnh của đồ thị.

Dữ liệu vào

- Dòng đầu chứa ba số nguyên dương $n \leq 10^4$, $m \leq 10^5$, $k \leq 10^4$;
- m dòng sau mỗi dòng chứa hai số nguyên dương u và v tương ứng với một cung (u, v) ;
- k dòng tiếp theo mỗi dòng chứa hai số nguyên dương i và j tương ứng với một thao tác thêm cung (i, j) vào đồ thị. Mỗi dòng là một yêu cầu truy vấn: sau thao tác thêm cung này yêu cầu chương trình của bạn trả lời số thành phần liên thông mạnh của đồ thị.

Kết quả

- Dòng đầu chứa duy nhất 1 số nguyên là số thành phần liên thông mạnh tìm được của đồ thị.
- k dòng tiếp theo mỗi dòng ghi ra số thành phần liên thông mạnh tìm được của đồ thị tương ứng với yêu cầu truy vấn trong file dữ liệu vào.

Ví dụ

<code>scc.inp</code>	<code>scc.out</code>
7 5 4	5
1 2	4
2 3	4
3 1	4
5 1	2
4 5	
3 5	
6 7	
7 4	
4 6	

Bài D. Bao đa giác

File dữ liệu vào: `polpack.inp`
File kết quả: `polpack.out`
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 256 MB

Cho một đa giác được tạo thành bởi n điểm (không nhất thiết phải lồi), nhiệm vụ của bạn là trả lời xem tồn tại một đường tròn với bán kính R chứa đa giác đó hay không.

Dữ liệu vào

File vào bao gồm nhiều test. Dòng đầu tiên của mỗi trường hợp là số n (với $n < 100$) là số đỉnh của đa giác. Tiếp theo là n dòng, mỗi dòng chứa 2 số thực xác định là tọa độ của các đỉnh. Dòng cuối cùng của mỗi trường hợp là bán kính R của đường tròn.

Kết thúc file là một số 0.

Dữ liệu đảm bảo không có một đỉnh xuất hiện 2 lần trong các test.

Kết quả

Với mỗi test:

Nếu tất cả các đỉnh đa giác nằm trong đường tròn với bán kính đã cho bạn in ra :

1

Nếu không thể thì in ra:

0

Ví dụ

<code>polpack.inp</code>	<code>polpack.out</code>
3 0 0 1 0 0 1 1.0 3 0 0 1 0 0 1 0.1 0	1 0

Bài E. Liệt kê xâu nhị phân

File dữ liệu vào: `binarylist.inp`
File kết quả: `binarylist.out`
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 256 MB

Cho 1 số nguyên dương n . Đưa ra xâu nhị phân độ dài n thứ k trong thứ tự từ điển mà không có i số 0 liên tiếp.

Dữ liệu vào

Dòng đầu ghi 3 số nguyên dương n, i, k .

Kết quả

Ghi ra xâu nhị phân độ dài n thứ k mà không có i số 0 liên tiếp trên một dòng duy nhất, các thành phần cách nhau bởi dấu cách. Nếu không tồn tại thì ghi ra -1.

Ví dụ

<code>binarylist.inp</code>	<code>binarylist.out</code>
6 4 2	0 1 1 0 1 0

Hạn chế

$n, i \leq 10^4$

- Subtask 1 : $k \leq 10^4$
- Subtask 2 : $k \leq 10^9$

Bài F. Bản vanxơ Fibonacci

File dữ liệu vào: `fibval.inp`
File kết quả: `fibval.out`
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 256 MB

Bản vanxơ Fibonacci là một bản nhạc mà giai điệu của nó bắt nguồn từ một trong những dãy số nổi tiếng nhất trong Lý thuyết số - dãy số Fibonacci. Hai số đầu tiên của dãy là số 1 và số 2, các số tiếp theo được xác định bằng tổng của hai số liên tiếp ngay trước nó trong dãy.

Bản vanxơ Fibonacci thu được bằng việc chuyển dãy số Fibonacci thành dãy các nốt nhạc theo qui tắc chuyển một số nguyên dương thành nốt nhạc sau đây:

- số 1 tương ứng với nốt Đô (C),
- số 2 tương ứng với nốt Rê (D),
- số 3 tương ứng với nốt Mi (E),
- số 4 tương ứng với nốt Fa (F),
- số 5 tương ứng với nốt Sol (G),
- số 6 tương ứng với nốt La (A),
- số 7 tương ứng với nốt Si (B),
- số 8 tương ứng với nốt Đô (C),
- số 9 tương ứng với nốt Rê (D)

và cứ tiếp tục như vậy. Ví dụ, dãy gồm 6 số Fibonacci đầu tiên 1, 2, 3, 5, 8 và 13 tương ứng với dãy các nốt nhạc C, D, E, G, C và A. Để xây dựng nhịp điệu vanxơ người ta đi tìm các đoạn nhạc có tính chu kỳ trong bản vanxơ Fibonacci. Đoạn nhạc được gọi là có tính chu kỳ nếu như có thể chia nó ra thành $k \geq 2$ đoạn giống hệt nhau. Ví dụ, đoạn nhạc GCAGCA là đoạn có tính chu kỳ, vì nó gồm hai đoạn giống nhau GCA. Yêu cầu: Cho trước hai số nguyên dương u, v ($u < v$), hãy xác định độ dài đoạn nhạc dài nhất có tính chu kỳ của bản nhạc gồm dãy các nốt nhạc của bản vanxơ Fibonacci bắt đầu từ vị trí u kết thúc ở vị trí v .

Dữ liệu vào

Dòng thứ nhất chứa số nguyên dương k ($k \leq 100$) là số lượng test;

Dòng thứ i trong số k dòng tiếp theo chứa hai số nguyên dương u_i, v_i được ghi cách nhau bởi dấu cách ($u_i < v_i \leq 10^9$) là vị trí bắt đầu và kết thúc của một bản nhạc;

Kết quả

Ghi ra k dòng, dòng thứ i chứa một số nguyên là độ dài đoạn nhạc tìm được tương ứng với test thứ i . Nếu không tìm được đoạn nào có tính chu kỳ thì ghi ra số -1.

Ví dụ

<code>fibval.inp</code>	<code>fibval.out</code>
2	-1
1 3	2
4 10	

Hạn chế

Ràng buộc: 50% số tests ứng với 50% số điểm của bài có $u_i < v_i \leq 100$.

Bài G. Chuỗi từ Fibonacci

File dữ liệu vào: FIBWORDS.inp
File kết quả: FIBWORDS.out
Hạn chế thời gian: 1 s
Hạn chế bộ nhớ: 256 MB

Chuỗi từ Fibonacci tạo bởi phép ghép các xâu nhị phân được định nghĩa như sau:

$$F(n) = \begin{cases} 0 & \text{nếu } n = 0 \\ 1 & \text{nếu } n = 1 \\ F(n-1) + F(n-2) & \text{nếu } n \geq 2 \end{cases}$$

Dưới đây là một số chuỗi từ Fibonacci đầu tiên:

n	F(n)
0	0
1	1
2	10
3	101
4	10110
5	10110101
6	1011010110110
7	101101011011010101
8	1011010110110101101011010110110
9	10110101101101011010110101101011010110101101011010110101

Yêu cầu: Cho trước một mẫu xâu p và một số n , hãy tính số lần xuất hiện của p trong $F(n)$.

Dữ liệu vào

Dữ liệu gồm nhiều test.

Dòng đầu tiên của mỗi test chứa một số nguyên n ($0 \leq n \leq 100$). Dòng thứ hai chứa xâu p . Dữ liệu đảm bảo xâu p không rỗng và có độ dài tối đa 100 000.

Kết quả

Với mỗi test, hiện thị số hiệu test và số lần xuất hiện của p trong $F(n)$ theo định dạng trong ví dụ. Các xâu xuất hiện có thể giao nhau. Dữ liệu đảm bảo số lần xuất hiện nhỏ hơn 2^{63} .

Ví dụ

FIBWORDS.inp	FIBWORDS.out
6	Case 1: 5
10	

Bài H. Thêm cạnh

File dữ liệu vào: `addedge.inp`
File kết quả: `addedge.out`
Hạn chế thời gian: 1 giây
Hạn chế bộ nhớ: 256 MB

Bình đang quan sát một đồ thị vô hướng. Bình bắn khoản liệu trong đồ thị này có tồn tại hai đỉnh nào mà sau khi nối hai đỉnh đó lại với nhau thì đồ thị sẽ có thêm duy nhất một chu trình đơn mới. Nhắc lại chu trình đơn là một chuỗi các đỉnh bắt đầu và kết thúc tại cùng một đỉnh mà không được phép lặp lại đỉnh ngoài đỉnh đầu và đỉnh cuối.

Yêu cầu: hãy giúp Bình tính số lượng cặp đỉnh thoả mãn điều kiện trên.

Dữ liệu vào

Dòng đầu tiên chứa hai số nguyên dương n, m ($n, m \leq 10^5$) tương ứng là số đỉnh và số cạnh của đồ thị. m dòng tiếp theo mỗi dòng chứa hai số nguyên dương u, v ($u, v \leq n$) là hai đỉnh của một cạnh.

Kết quả

Ghi ra duy nhất một số là số cặp đỉnh tìm được.

Ví dụ

<code>addedge.inp</code>	<code>addedge.out</code>
5 4 1 2 2 3 3 4 4 5	6
5 5 1 2 2 3 1 3 3 4 4 5	1