

Mục lục

Liệt kê hoán vị - PERMUTATIONLIST	1
Bài toán người du lịch - TSP	2
Kế hoạch định tuyến - ROUTING	3
Lấy vàng — GOLD	4
Xếp lịch thí nghiệm — MACHINE	5
Trò chơi xổ số – LOTTO	6
Bản vanxơ Fibonacci — FIBVAL	7
Liệt kê xâu nhị phân - BINARYLIST	8
Chuỗi từ Fibonacci — FIBWORDS	9
Biển diễn đồ thị - GREP	10
Thành phần liên thông mạnh - SCC	11
Thêm cạnh - ADDEDGE	12
Sắp xếp hạn chế bộ nhớ (Obama sort) — SORT3MB	13
Đường đi ngắn nhất - MINPATH	14
Cây khung nhỏ nhất - MST	15
Tìm kiếm trên đồ thị - GSEARCH	16
Biển diễn đồ thị - GREP	17
Phương pháp sinh xâu nhị phân - BINARYGEN	18
Phương pháp sinh chuỗi tổ hợp - COMBINATIONGEN	19
Phương pháp sinh hoán vị - PERMUTATIONGEN	20
Bài toán cái túi - KNAPSAC	21
Liệt kê tổ hợp - COMBINATIONLIST	22

Bài A. 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 $n, k \leq 10^4$

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

Bài B. Bài toán người du lịch

File dữ liệu vào: TSP.inp
File kết quả: TSP.out
Hạn chế thời gian: 0.2 giây
Hạn chế bộ nhớ: 256 MB

Một người du lịch xuất phát từ thành phố thứ nhất muốn đi thăm quan tất cả $n - 1$ thành phố khác. mỗi thành phố đứng một lần, rồi quay trở lại thành phố xuất phát.

Yêu cầu: Cho biết chi phí đi lại giữa các thành phố, hãy giúp người du lịch tìm hành trình với tổng chi phí là nhỏ nhất.

Dữ liệu vào

Dòng đầu tiên chứa hai số nguyên dương n, m cách nhau bởi dấu cách ($n \leq 20, m < 400$).

m dòng tiếp theo mỗi dòng chứa ba số nguyên dương i, j, c ($i, j \leq n, c \leq 10^6$) biểu thị chi phí đi trực tiếp từ thành phố i đến thành phố j là c .

Lưu ý: nếu từ thành phố i đến thành phố j nào không mô tả chi phí đi lại thì có nghĩa là không có đường đi trực tiếp từ i đến j .

Kết quả

Ghi ra duy nhất một số là tổng chi phí hành trình nhỏ nhất tìm được.

Ví dụ

TSP.inp	TSP.out
2 2 1 2 3 2 1 2	5

Bài C. Kế hoạch định tuyến

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

Đảo Alpha là một hòn đảo đẹp như tranh vẽ và chỉ cho phép phương tiện giao thông duy nhất là xe đạp trên đảo. Do đó, có nhiều tuyến đường xe đạp kết nối các điểm chụp ảnh đẹp nhất trên đảo. Để giúp du khách có kế hoạch chuyến đi của họ tới hòn đảo này, ủy ban du lịch muốn thực hiện định tuyến cho đường xe đạp mà đi qua một số các điểm chụp ảnh đẹp nhất trên đảo.

Yêu cầu: Cho một bản đồ của tất cả những con đường xe đạp trên đảo và một danh sách những địa điểm chụp ảnh đẹp nhất trên mỗi tuyến đường quy hoạch (những điểm không được đề cập thì không xuất hiện trên tuyến đường đó), hãy giúp ủy ban du lịch lên kế hoạch định tuyến từng tuyến đường r để khoảng cách trên mỗi con đường là tối thiểu. Lưu ý rằng mỗi điểm chụp ảnh đẹp nhất chỉ có thể xuất hiện nhiều nhất một lần trên các tuyến đường.

Dữ liệu vào

Gồm hai phần. Phần đầu tiên cung cấp các thông tin của những tuyến đường xe đạp trên đảo. Dòng đầu tiên chứa hai số nguyên $n \leq 100$ và $r \leq 10$ là n điểm chụp ảnh đẹp nhất trên đảo và có r tuyến đường được lên kế hoạch.

n dòng tiếp theo (dòng 2 đến dòng thứ $n + 1$) chứa $n \times n$ số nguyên trong khoảng từ 0 đến 10 (n dòng với n số nguyên trên mỗi dòng), số nguyên thứ j trên dòng i là khoảng cách từ chỗ chụp ảnh đẹp nhất $i - 1$ đến chỗ chụp ảnh đẹp nhất j ; giá trị bằng 0 nghĩa là không có con đường một chiều đi từ chỗ chụp ảnh đẹp nhất $i - 1$ đến chỗ chụp ảnh đẹp nhất j .

Phần thứ hai gồm r dòng biểu thị r tuyến đường tham quan cần được quy hoạch. Mỗi dòng liệt kê các điểm chụp ảnh đẹp nhất cần đưa vào con đường tương ứng. Các số nguyên trên mỗi dòng biểu thị các địa điểm được đề nghị là các điểm chụp ảnh đẹp nhất trên tuyến đường tương ứng. Số nguyên đầu tiên trên dòng là điểm bắt đầu của tuyến đường và số nguyên cuối cùng là điểm dừng cuối cùng trên tuyến. Tuy nhiên, các điểm dừng ở giữa có thể được thăm theo thứ tự bất kì.

Kết quả

Đưa ra r số nguyên trên r dòng (mỗi số nguyên trên một dòng) là khoảng cách của từng tuyến đường được quy hoạch. Nếu có một tuyến đường không thể quy hoạch được thì in ra 0.

Ví dụ

<code>routing.inp</code>	<code>routing.out</code>
6 3	5
0 1 2 0 1 1	0
1 0 1 1 1 0	7
0 2 0 1 3 0	
4 3 1 0 0 0	
0 0 1 1 0 0	
1 0 0 0 0 0	
1 3 5	
6 3 2 5	
6 1 2 3 4 5	

Bài D. Lầy vàng

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

Ông vua ở một vương quốc nọ có n kho vàng nằm trên một đường thẳng được đánh số theo thứ tự là $1, 2, \dots, n$, trong đó kho vàng i có trữ lượng vàng là a_i (a_i là số nguyên không âm) và đặt tại toạ độ i ($\forall i = 1, \dots, n$). Nhà vua mở cuộc thi tài cho những người tìm vàng xem ai là người có thể lựa chọn được dãy các kho vàng có tổng trữ lượng lớn nhất với điều kiện khoảng cách giữa 2 kho liên tiếp trong dãy các kho được chọn ra phải lớn hơn hoặc bằng L_1 và nhỏ hơn hoặc bằng L_2 , nghĩa là, nếu dãy kho được chọn là i_1, i_2, \dots, i_k ($i_1 < i_2 < \dots < i_k$), thì ta phải có $L_1 \leq i_j - i_{j-1} \leq L_2$, $j = 2, 3, \dots, k$.

Dữ liệu vào

Dòng 1 ghi giá trị n, L_1, L_2 cách nhau bởi dấu cách ($1 \leq n \leq 10^6, 1 \leq L_1 \leq L_2 \leq n$)

Dòng 2 ghi n giá trị nguyên a_1, a_2, \dots, a_n cách nhau bởi dấu cách.

Kết quả

Ghi ra duy nhất một số là tổng trữ lượng của các kho vàng được lựa chọn.

Ví dụ

gold.inp	gold.out
6 2 2 3 5 9 6 7 4	19

Giải thích

Phương án tối ưu là lựa chọn dãy 3 kho vàng 1, 3 và 5 với tổng dự trữ là $3 + 9 + 7 = 19$

Bài E. 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 F. Trò chơi xổ số

File dữ liệu vào: lotto.inp
File kết quả: lotto.out
Hạn chế thời gian: 0.1s
Hạn chế bộ nhớ: 512 MB

Trong một trò chơi xổ số của Đức bạn phải chọn 6 con số từ tập $\{1, 2, \dots, 49\}$. Một chiến lược phổ biến để chơi xổ số – mặc dù nó không làm tăng cơ hội chiến thắng của bạn – là để chọn một tập con S chứa k ($k > 6$) của 49 con số này, và sau đó chơi một số ván với lựa chọn các số chỉ trong S .

Ví dụ, đối với $k = 8$ và $S = \{1, 2, 3, 5, 8, 13, 21, 34\}$ có thể có 28 ván: $[1, 2, 3, 5, 8, 13]$, $[1, 2, 3, 5, 8, 21]$, $[1, 2, 3, 5, 8, 34]$, $[1, 2, 3, 5, 13, 21]$, \dots , $[3, 5, 8, 13, 21, 34]$.

Yêu cầu: Viết chương trình đọc số k và tập S , sau đó in tất cả các ván có thể có với các số được chọn chỉ trong S .

Dữ liệu vào

Bao gồm duy nhất một dòng chứa vài số nguyên ngăn cách với nhau bằng dấu cách. Số nguyên đầu tiên trên dòng sẽ là số k ($6 < k < 13$). k số nguyên tiếp theo là các số trong tập S có thể theo thứ tự tăng dần hoặc không.

Kết quả

In ra tất cả các ván chơi có thể có, mỗi ván trên một dòng.

Các con số của mỗi ván chơi phải được sắp xếp theo thứ tự tăng dần, ngăn cách nhau bởi đúng một dấu cách. Các ván chơi phải được sắp xếp thứ tự từ điển, có nghĩa là ván có thứ tự từ điển thấp nhất ghi trên dòng đầu tiên, sau đó là thấp thứ hai, \dots , như trong ví dụ đầu ra dưới đây.

Ví dụ

lotto.inp	lotto.out
7 1 2 3 4 5 6 7	1 2 3 4 5 6 1 2 3 4 5 7 1 2 3 4 6 7 1 2 3 5 6 7 1 2 4 5 6 7 1 3 4 5 6 7 2 3 4 5 6 7
7 1 2 4 3 5 7 6	1 2 3 4 5 6 1 2 3 4 5 7 1 2 3 4 6 7 1 2 3 5 6 7 1 2 4 5 6 7 1 3 4 5 6 7 2 3 4 5 6 7

Bài G. Bản vanxơ Fibonacci

File dữ liệu vào: `fibval.inp`
File kết quả: `fibval.out`
Hạn chế thời gian: 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 H. Liệt kê xâu nhị phân

File dữ liệu vào: `binarylist.inp`
File kết quả: `binarylist.out`
Hạn chế thời gian: 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 $n, k, i \leq 10^4$

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 I. 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	10110101101101010101
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 J. Biểu diễn đồ thị

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

Bài toán yêu cầu chuyển đổi giữa các cách biểu diễn đồ thị và kiểm tra các đỉnh kề.

Dữ liệu vào

- Dòng đầu chứa ba số nguyên dương $n \leq 10^5$, $m \leq 10^5$ và $k \leq 10^5$;
- m dòng sau mỗi dòng chứa hai số nguyên dương u và v tương ứng với một cung (u, v) ;
- Dòng tiếp theo chứa k số nguyên dương w tương ứng với k yêu cầu liệt kê các đỉnh đầu của cung có đỉnh cuối là w theo thứ tự tăng dần chỉ số đỉnh.

Kết quả

- Dòng đầu chứa hai số n và m ;
- Dòng 2 chứa mảng `Head` gồm n giá trị: `Head[i]` là vị trí đầu của khoảng chứa thông tin các đỉnh kề với i trong mảng danh sách kề `Last`. Nếu đỉnh i không có đỉnh kề thì `Head[i]=Head[i + 1]`;
- Dòng 3 in ra mảng `Last` gồm m số là thông tin các đỉnh kề trong danh sách kề;
- k dòng tiếp theo mỗi dòng in các đỉnh có chỉ số tăng dần là các đỉnh đầu của cung có đỉnh cuối là w tương ứng với yêu cầu trong file dữ liệu vào. Nếu không có cung đến w thì để dòng trống.

Ví dụ

<code>grep.inp</code>	<code>grep.out</code>
4 5 3	4 5
1 2	1 3 5 6
3 4	2 4 3 4 4
2 3	[Dòng trống]
2 4	2
1 4	1 2 3
1 3 4	

Bài K. 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 $n \leq 10^5$, $m \leq 10^5$, $k \leq 10^5$;
- 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 L. Thêm cạnh

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

Bình đang quan sát một đồ thị vô hướng. Bình bản 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>adddge.inp</code>	<code>adddge.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

Bài M. Sắp xếp hạn chế bộ nhớ (Obama Sort)

File dữ liệu vào: `sort3mb.inp`
File kết quả: `sort3mb.out`
Hạn chế thời gian: 1 giây
Hạn chế bộ nhớ: 3 MB

Tổng thống Obama trong một lần thăm trung tâm tuyển trạch của Microsoft, vị giám đốc tuyển trạch hỏi Obama một câu tuyển dụng như sau:

“Theo tổng thống thì sắp xếp 1 triệu số nguyên 32 bit thì thuật toán nào chạy nhanh nhất?”

Theo các bạn thì:

- Câu trả lời đúng là gì?
- Obama sẽ trả lời như thế nào?

Obama không phải là người làm tin học, tuy nhiên ông đã có một câu trả lời rất thông minh:

“...<câu trả lời của Obama có liên quan đến Bubble Sort> ...”

Cần bình luận thêm là từ Bubble Sort có thể là một từ thông dụng trong câu nói của người nói tiếng Anh khi nói đến cách sắp xếp các đồ vật hay công việc. Tư tưởng rất tự nhiên như sau: cứ nhìn vào hai đồ vật liên tiếp nếu thấy không đúng vị trí thì đảo chỗ chúng; cứ làm như vậy cho đến khi không còn cặp nào như vậy.

Câu hỏi thêm: Chứng minh tính đúng đắn của Bubble Sort.

Từ câu chuyện vui, các bạn hãy làm bài có tên là Obama Sort (Sort3MB) mở rộng cho số thực và hạn chế bộ nhớ tối đa 3MB.

Dữ liệu vào

Dòng đầu chứa số nguyên $n \leq 10^6$.

Dòng 2 chứa n số thực giá trị tối đa 4 bytes và có 2 chữ số sau dấu phẩy động.

Kết quả

Một dòng duy nhất chứa n số đã được sắp xếp tăng dần với 2 chữ số sau dấu phẩy động. Lưu ý là giá trị các số trong file input kể cả phần lẻ cũng phải giống hệt như giá trị tương ứng trong dữ liệu đầu vào.

Ví dụ

<code>sort3mb.inp</code>	<code>sort3mb.out</code>
6 2.22 5.25 6.26 1.21 4.24 3.23	1.21 2.22 3.23 4.24 5.25 6.26

Hạn chế

- 50% số test mà tất cả các giá trị là số nguyên (.00)
- 50% số test có $n \leq 750000$

Bài N. Đường đi ngắn nhất

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

Bài toán yêu cầu trong số các đường đi ngắn nhất trên một đồ thị hãy tìm và đưa ra đường đi có thứ tự từ điển nhỏ nhất.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương $n \leq 10^5$, $m \leq 10^5$ là số đỉnh và số cạnh của đồ thị;
- m dòng sau mỗi dòng chứa ba số nguyên dương u , v và c tương ứng là một cung (u, v) và trọng số c trên cung đó;
- Dòng cuối cùng chứa hai số nguyên dương s và t và đỉnh đầu và cuối của đường đi cần tìm.

Kết quả

- Dòng đầu tiên chứa 1 số nguyên là độ dài của đường đi tìm được.
- Dòng thứ 2 chứa các số nguyên là các đỉnh của đường đi tìm được.

Ví dụ

<code>minpath.inp</code>	<code>minpath.out</code>
4 6 2 1 3 2 3 1 3 1 10 4 1 5 3 4 5 4 2 6 2 4	6 2 3 4

Bài O. Cây khung nhỏ nhất

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

Bài toán yêu cầu trong số các cây khung nhỏ nhất của một đồ thị vô hướng hãy tìm và đưa ra cây khung có thứ tự từ điển nhỏ nhất.

Định nghĩa thứ tự từ điển của 1 cây khung:

- Cây khung được biểu diễn bởi chuỗi các cạnh của cây khung đó
- Mỗi cạnh được viết theo chỉ số nhỏ đến chỉ số lớn: (u, v) , $u < v$
- Các cạnh của cây khung được liệt kê từ cạnh có chỉ số nhỏ đến chỉ số lớn, cạnh (u_1, v_1) đi trước cạnh (u_2, v_2) nếu $u_1 < u_2$ hoặc $(u_1 = u_2$ và $v_1 < v_2)$.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương $n \leq 10^5$, $m \leq 10^5$ là số cạnh và số đỉnh của đồ thị;
- m dòng sau mỗi dòng chứa ba số nguyên dương u , v và c tương ứng là một cạnh (u, v) và trọng số c trên cạnh;

Kết quả

- Dòng đầu tiên chứa 1 số nguyên là trọng số của cây khung tìm được.
- Dòng thứ 2 chứa $n - 1$ cặp số là các cạnh của cây khung tìm được, liệt kê theo thứ tự từ chỉ số nhỏ đến chỉ số lớn của từng cạnh.

Ví dụ

<code>mst.inp</code>	<code>mst.out</code>

Bài P. Tìm kiếm trên đồ thị

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

Bài toán yêu cầu kiểm tra tính liên thông của đồ thị và tìm đường đi giữa hai đỉnh i và j cho trước.

Yêu cầu 1: Tính số thành phần liên thông của đồ thị.

Yêu cầu 2: Tìm đường đi qua ít cạnh nhất từ i đến j .

Yêu cầu 3: Tìm đường đi có thứ tự từ điển nhỏ nhất từ i đến j .

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương $n \leq 10^5$, $m \leq 10^5$;
- m dòng sau mỗi dòng chứa hai số nguyên dương u và v tương ứng với một cung (u, v) ;
- Dòng tiếp theo chứa 2 số nguyên dương i và j tương ứng với yêu cầu tìm đường đi giữa hai đỉnh i và j .

Kết quả

- Dòng đầu tiên chứa 1 số nguyên là số thành phần liên thông của đồ thị tìm được.
- Dòng thứ 2 chứa một số nguyên là độ dài đường đi tìm được trong yêu cầu 2.
- Dòng thứ 3 chứa một số nguyên ℓ là độ dài đường đi tìm được trong yêu cầu 3.
- Dòng thứ 4 chứa ℓ số nguyên là chỉ số các đỉnh trên đường đi tìm được của yêu cầu 3 lần lượt từ đỉnh i kết thúc tại đỉnh j .

Ví dụ

<code>gsearch.inp</code>	<code>gsearch.out</code>

Bài Q. Biểu diễn đồ thị

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

Bài toán yêu cầu chuyển đổi giữa các cách biểu diễn đồ thị và kiểm tra các đỉnh kề.

Dữ liệu vào

- Dòng đầu chứa ba số nguyên dương $n \leq 10^5$, $m \leq 10^5$ và $k \leq 10^5$;
- m dòng sau mỗi dòng chứa hai số nguyên dương u và v tương ứng với một cung (u, v) ;
- Dòng tiếp theo chứa k số nguyên dương w tương ứng với k yêu cầu liệt kê các đỉnh đầu của cung có đỉnh cuối là w theo thứ tự tăng dần chỉ số đỉnh.

Kết quả

- Dòng đầu chứa hai số n và m ;
- Dòng 2 chứa mảng `Head` gồm n giá trị: `Head[i]` là vị trí đầu của khoảng chứa thông tin các đỉnh kề với i trong mảng danh sách kề `Last`. Nếu đỉnh i không có đỉnh kề thì `Head[i]=Head[i + 1]`;
- Dòng 3 in ra mảng `Last` gồm m số là thông tin các đỉnh kề trong danh sách kề;
- k dòng tiếp theo mỗi dòng in các đỉnh có chỉ số tăng dần là các đỉnh đầu của cung có đỉnh cuối là w tương ứng với yêu cầu trong file dữ liệu vào. Nếu không có cung đến w thì để dòng trống.

Ví dụ

<code>grep.inp</code>	<code>grep.out</code>
4 5 3	4 5
1 2	1 3 5 6
3 4	2 4 3 4 4
2 3	[Dòng trống]
2 4	2
1 4	1 2 3
1 3 4	

Bài R. Phương pháp sinh xâu nhị phân

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

Cho 1 một xâu nhị phân S độ dài n . Tìm xâu nhị phân kế tiếp của xâu S trong thứ tự từ điển.

Dữ liệu vào

20 test

Dòng đầu 1 số nguyên dương $n \leq 10^4$ Dòng thứ 2 ghi n số 0 hoặc 1 liên tiếp nhau.

Kết quả

Ghi ra xâu nhị phân kế tiếp của xâu S trên một dòng duy nhất. Nếu không tồn tại thì ghi ra -1.

Ví dụ

<code>np.inp</code>	<code>np.out</code>

Bài S. Phương pháp sinh chuỗi tổ hợp

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

Cho 1 một chuỗi tổ hợp C độ dài m với các thành phần nằm trong tập $\{1, 2, \dots, n\}$. Tìm chuỗi tổ hợp kế tiếp của chuỗi C trong thứ tự từ điển.

Dữ liệu vào

20 test

Dòng đầu 2 số nguyên dương $n, m \leq 10^4$ Dòng thứ 2 ghi m số nguyên dương $\leq n$ cách nhau bởi dấu cách.

Kết quả

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

Ví dụ

<code>th.inp</code>	<code>th.out</code>

Bài T. Phương pháp sinh hoán vị

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

Cho 1 một hoán vị H độ dài n với các thành phần nằm trong tập $\{1, 2, \dots, n\}$. Tìm hoán vị kế tiếp của hoán vị H trong thứ tự từ điển.

Dữ liệu vào

20 test

Dòng đầu ghi 1 số nguyên dương $n \leq 10^4$ Dòng thứ 2 ghi n số nguyên dương $\leq n$ cách nhau bởi dấu cách là hoán vị H .

Kết quả

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

Ví dụ

<code>hv.inp</code>	<code>hv.out</code>

Bài U. Bài toán cái túi

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

Một nhà thám hiểm cần đem theo một cái túi có trọng lượng không quá b . Có n đồ vật có thể đem theo. Đồ vật thứ j có trọng lượng a_j và giá trị sử dụng c_j . Hỏi nhà thám hiểm cần đem theo những đồ vật nào để cho tổng giá trị sử dụng là lớn nhất mà tổng trọng lượng đồ vật mang theo cái túi không vượt quá b ?

Dữ liệu vào

Dòng đầu tiên chứa hai số nguyên dương n, b ($n \leq 30, b \leq 10^6$).

Dòng thứ j trong số n dòng tiếp theo mỗi dòng ghi ra hai số nguyên dương $a_j, c_j \leq 10^6$.

Kết quả

Ghi ra duy nhất một số là tổng giá trị lớn nhất tìm được của các đồ vật cho vào túi.

Ví dụ

<code>knapsac.inp</code>	<code>knapsac.out</code>

Bài V. Liệt kê tổ hợp

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây
Hạn chế bộ nhớ: 256 MB

Cho 2 số nguyên dương n, m . Đưa ra chuỗi tổ hợp chập m của n phần tử trong tập $\{1, 2, \dots, n\}$ thứ k trong thứ tự từ điển.

Dữ liệu vào

20 test

Dòng đầu ghi 3 số nguyên dương $n, m, k \leq 10^4$

Kết quả

Ghi ra chuỗi tổ hợp chập m của n phần tử 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

stdin	stdout