Mục lục

Bài A. Phương pháp sinh xâu nhị phân

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	0.1 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

stdin	stdout
5 00100	00101

Bài B. Phương pháp sinh chuỗi tổ hợp

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	0.1 giây
Hạn chế bộ nhớ:	256 MB

Cho 1 một chuỗi tổ hợp C độ dài m với các thành phần nằm trong tập $\{1, 2, \ldots, n\}$. 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

stdin	stdout
5 3 2 3 5	2 4 5

Bài C. Phương pháp sinh hoán vị

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
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 $\{1, 2, \ldots, n\}$. 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

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

Bài D. Liệt kê xâu nhị phân

File dữ liệu vào:	`stdinp`
File kết quả:	`stdout`
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 $n, k, i \leq 10^4$ cách nhau bởi dấu cách

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

stdinp	stdout
6 4 2	0 1 1 0 1 0

Bài E. Liệt kê tổ hợp

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	0.1 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 cách nhau bởi dấu cách, $n, m \leq 10^4; k \leq 10^9$.

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
7 3 6	1 3 4

Bài F. Liệt kê hoán vị

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
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

20 test

Dòng đầu ghi 2 số nguyên dương n, k ($n \leq 10^4, k \leq 10^9$) cách nhau bởi dấu cách.

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

stdin	stdout
3 4	2 3 1

Bài G. 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 đúng 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 hai 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 H. 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

knapsac.inp	knapsac.out

Bài I. 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, các đỉnh kề của một đỉnh phải được đưa ra theo thứ tự tăng dần của chỉ số đỉnh;
- 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

grep.inp	grep.out
4 5 3 1 2 3 4 2 3 2 4 1 4 1 3 4	4 5 1 3 5 6 2 4 3 4 4 2 1 2 3

Bài J. 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

gsearch.inp	gsearch.out

Bài K. Đường đi ngắn nhất

File dữ liệu vào:	`minpath.inp`
File kết quả:	`minpath.out`
Hạn chế thời gian:	0.1 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

minpath.inp	minpath.out
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 L. 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 canh (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

mst.inp	mst.out

Bài M. 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

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