

Mục lục

Sắp xếp hạn chế bộ nhớ — SORT3MB	1
Đoán Cấu Trúc Dữ Liệu — GuessDST	2
Lỗ Ban - HIST	4
Dãy con cực đại với độ dài bị chặn - MAXSUBSEQ	5
Danh sách móc nối — LINKLIST	6

Bài A. Sắp xếp hạn chế bộ nhớ

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

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.

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

Bài B. Đoán Cấu Trúc Dữ Liệu

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

Hãy tưởng tượng bạn có một cấu trúc dữ liệu dạng chiếc túi với hai chức năng:

- 1 x : Đưa phần tử x vào túi.
- 2 : Lấy một phần tử ra khỏi túi.

Sau một chuỗi thao tác với chiếc túi, bạn sẽ phải đoán cấu trúc dữ liệu của nó. Liệu nó sẽ là ngăn xếp (stack, vào sau ra trước), hàng đợi (queue, vào trước ra trước), hàng đợi ưu tiên (priority-queue, luôn luôn lấy phần tử lớn nhất trước), hay thứ gì khác mà bạn khó có thể tưởng tượng?

Dữ liệu vào

Dòng đầu chứa số nguyên n trong đoạn từ 1 đến 100. Mỗi dòng trong n dòng tiếp theo là một câu lệnh loại 1, hoặc số 2 cùng một số nguyên x (với ý nghĩa khi thực hiện câu lệnh loại 2 sẽ trả lại phần tử x). Giá trị của x luôn là số dương không lớn hơn 100.

Kết quả

Đưa ra một trong các kết quả tùy thuộc vào cấu trúc dữ liệu của chiếc túi:

`stack`

Thao tác dạng ngăn xếp

`queue`

Thao tác dạng hàng đợi

`priority queue`

Thao tác dạng hàng đợi ưu tiên

`impossible`

Không phải một trong ba dạng trên

`not sure`

Có thể cùng là nhiều hơn một trong ba dạng trên

Ví dụ

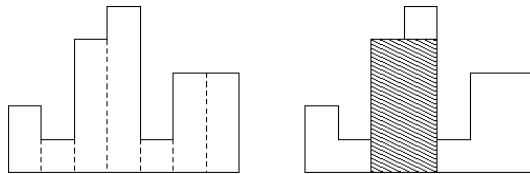
guessDST.inp	guessDST.out
6 1 1 1 2 1 3 2 1 2 2 2 3	queue
6 1 1 1 2 1 3 2 3 2 2 2 1	not sure
2 1 1 2 2	impossible

Bài C. Lỗ Ban

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

Lỗ Ban là một vị thợ mộc nổi tiếng bậc nhất thời Tống của Trung Quốc với đôi bàn tay cực kỳ tài hoa khéo léo. Ngưỡng mộ tài năng của Lỗ Ban, Vua Tống mời Lỗ Ban vào triều làm quan chuyên quản lý việc thiết kế cung điện và chế tác vật dụng tinh xảo. Vua Tống trong một chuyến du ngoạn ở núi Ngũ Nhạc tình cờ phát hiện một phiến đá ngũ sắc tuyệt đẹp, Vua Tống nảy ra ý định đem phiến đá này về cung để chế tác thành một bàn cờ. Nhưng Vua Tống nhanh chóng phát hiện ra rằng, phiến đá này có hình dạng kỳ lạ rất khó để có thể cắt ra phần diện tích vuông vắn đủ lớn cho bàn cờ. Vậy là Vua Tống triệu Lỗ Ban vào triều để thương lượng:

Bề mặt phiến đá có thể được mô tả như là một hình đa giác được ghép thành từ nhiều phiến đá nhỏ hình chữ nhật có chung nhau một mép, có các chiều dài khác nhau nhưng giống nhau về chiều rộng và bằng 1 đơn vị. Trong hình vẽ dưới đây, phiến đá đa giác gồm các hình chữ nhật có chiều cao lần lượt từ trái qua phải là 2,1,4,5,1,3,3 và chiều rộng đều bằng 1.



Yêu cầu: Bạn cần giúp Lỗ Ban tìm ra hình chữ nhật chung mép với các hình chữ nhật nhỏ và có diện tích lớn nhất nằm trong phiến đá đa giác nói trên. Ở hình vẽ dưới, hình chữ nhật lớn nhất là hình được gạch chéo.

Dữ liệu vào

Chứa một hoặc nhiều test. Mỗi test mô tả một đa giác bắt đầu bằng số nguyên n ($1 \leq n \leq 1000000$) là số lượng hình chữ nhật nhỏ cấu thành đa giác. Tiếp theo sau là n số nguyên l_1, l_2, \dots, l_n với $0 \leq l_i \leq 100000000$ lần lượt từ trái sang phải biểu thị chiều dài của các hình chữ nhật. Chiều rộng của các hình chữ nhật bằng nhau và bằng 1. File kết thúc với dòng ghi duy nhất một số 0.

Kết quả

Với mỗi test ghi trên một dòng diện tích của hình chữ nhật nằm trong đa giác thỏa mãn điều kiện đề bài.

Ví dụ

<code>hist.inp</code>	<code>hist.out</code>
7 2 1 4 5 1 3 3	8
4 1000 1000 1000 1000	4000
0	

Bài D. Dãy con cực đại với độ dài bị chặn

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

Cho dãy số nguyên $A = a_1, a_2, \dots, a_n$. Ta gọi dãy con của dãy A là dãy gồm một số phần tử liên tiếp của nó, tức là dãy có dạng a_i, a_{i+1}, \dots, a_j ($1 \leq i \leq j \leq n$). Ta gọi độ dài của dãy là số lượng phần tử của nó còn trọng số của dãy là tổng các phần tử của nó. Một dãy con được gọi là có độ dài bị chặn nếu nó có độ dài lớn hơn hoặc bằng L_1 và nhỏ hơn hoặc bằng L_2 .

Yêu cầu: Cho dãy gồm n số nguyên a_1, a_2, \dots, a_n và hai số L_1 và L_2 , cần tìm trọng số của dãy con có độ dài bị chặn với trọng số lớn nhất.

Dữ liệu vào

Dòng 1 chứa 4 số nguyên n, L_1, L_2 được ghi cách nhau bởi dấu cách ($1 \leq n \leq 10^6, 1 \leq L_1 \leq L_2 \leq n$);

Dòng 2 chứa n số nguyên a_1, a_2, \dots, a_n được ghi cách nhau bởi dấu cách là các phần tử của dãy số.

Kết quả

Ghi ra duy nhất một số nguyên là trọng số tìm được.

Ví dụ

maxsubseq.inp	maxsubseq.out
6 3 4 3 5 -9 6 7 -4	9

Dãy

con độ dài hạn chế có trọng số lớn nhất là dãy 5, -9, 6, 7 với trọng số bằng 9.

Bài E. Danh sách móc nối

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

0	1	2	3	4	5	6	7	8	9
2	5	4	4	-1	1	-1	3	0	8

Hàng dưới bảng trên chứa thông tin của mảng A kèm các chỉ số mảng ở hàng trên. Mảng A chứa các con trỏ của một danh sách móc nối tuyến tính. Trong bài toán này, một con trỏ đơn giản chỉ là một giá trị nguyên. Con trỏ tại nốt đầu tiên được lưu tại $A[0]$, có nghĩa là giá trị $A[0]$ chỉ ra vị trí của nốt thứ hai. Con trỏ của nốt thứ hai được lưu tại $A[A[0]]$, rồi con trỏ của nốt thứ ba được lưu tại $A[A[A[0]]]$, Con trỏ có giá trị -1 biểu thị điểm cuối của danh sách móc nối.

Trong ví dụ trên, giá trị của $A[0]$ là 2 nghĩa là con trỏ thứ hai được lưu tại $A[2]$. Giá trị của $A[2]$ là 4 nghĩa là con trỏ thứ 3 được lưu tại $A[4]$. Giá trị của $A[4]$ là -1 nghĩa là không còn nốt nối sau đó nữa. Như vậy chuỗi con trỏ trong danh sách móc nối trên sẽ gồm 3 nốt như sau: $A[0]=2 \rightarrow A[2]=4 \rightarrow A[4]=-1$.

Cho trước các giá trị của mảng con trỏ A và biết rằng có một vị trí đã bị thay đổi giá trị mới nhưng không được biết thông tin về vị trí và giá trị mới đó. Lưu ý là giá trị mới có thể bằng giá trị cũ, nghĩa là bản chất mảng không bị thay đổi. Đây là một ứng dụng trong hình sự để truy vết tội phạm. Vì vậy để khôi phục lại mảng ban đầu thì cần sửa giá trị tại đúng một vị trí trong mảng mới sao cho mảng được sửa biểu diễn một danh sách móc nối có nhiều nốt nhất. Với ví dụ trên:

- Đổi giá trị $A[4]$ thành 6 sẽ tạo ra danh sách móc nối với 4 nốt.
- Đổi giá trị $A[0]$ thành 7 sẽ tạo ra danh sách móc nối với 4 nốt.
- Đổi giá trị $A[0]$ thành 9 sẽ tạo ra danh sách móc nối sai.
- Đổi giá trị $A[2]$ thành 7 sẽ tạo ra danh sách móc nối với 5 nốt.

Trong số tất cả các cách đổi có thể, kể cả cách không đổi gì, thì danh sách móc nối mới tạo ra với 5 nốt là dài nhất. Do vậy nhiều khả năng giá trị ban đầu của $A[2]$ là 7.

Yêu cầu: tìm cách thay đổi giá trị một nốt để có được mảng biểu diễn danh sách móc nối dài nhất.

Dữ liệu vào

Dòng đầu tiên chứa duy nhất một số nguyên N là kích thước của mảng A .

Dòng thứ hai ghi ra N số nguyên là các giá trị trong mảng, bắt đầu từ $A[0]$. Mỗi số nguyên ≥ -1 và $< N$.

Kết quả

Ghi ra duy nhất một số là số nốt của danh sách móc nối có nhiều nốt nhất tìm được.

Ví dụ

<code>linklist.inp</code>	<code>linklist.out</code>
10 2 5 4 4 -1 1 -1 3 0 8	5

Hạn chế

- Subtask 1 (20%): $1 < N \leq 20$
- Subtask 2 (20%): $20 < N \leq 3000$ và số lượng nốt tối đa của danh sách móc nối tìm được không quá 100.
- Subtask 3 (20%): $20 < N \leq 3000$
- Subtask 4 (40%): $3000 < N \leq 20000$