

PHỦ

Cho n đoạn trên trục số, đoạn thứ i là $[l_i, h_i]$. Hãy chọn ra trong các đoạn kể trên một số ít nhất các đoạn để phủ hết đoạn $[a, b]$.

Dữ liệu: Vào từ file văn bản COVER.INP

- Dòng 1: Chứa 3 số n, a, b
- n dòng tiếp theo, dòng thứ i chứa hai số l_i, h_i

Kết quả: Ghi ra file văn bản COVER.OUT

- Dòng 1: Ghi số k là số đoạn được chọn (Nếu không có cách chọn thì $k = -1$)
- Trong trường hợp có phương án thực hiện yêu cầu thì k dòng tiếp theo, mỗi dòng ghi chỉ số một đoạn được chọn

Các số trên một dòng của Input/Output files được/phải ghi cách nhau ít nhất một dấu cách

Ràng buộc: Các số trong Input File là số nguyên dương $\leq 10^5$; $a \leq b$; $\forall i: l_i \leq h_i$

Ví dụ:

COVER. INP	COVER. OUT
8 2 10	3
4 8	1
1 3	4
2 3	6
1 4	
3 4	
7 10	
9 11	
8 11	

COVER. INP	COVER. OUT
4 1 200	-1
1 4	
2 5	
4 5	
6 45	

ĐẠI DIỆN

Trên trục số cho n đoạn đóng, đoạn thứ i là $[L_i, R_i]$. Hãy chọn ra một tập ít nhất các điểm nguyên phân biệt trên trục số thoả mãn: Mỗi đoạn trong số n đoạn đã cho phải chứa tối thiểu 2 điểm trong tập này.

Dữ liệu: Vào từ file văn bản REP.INP

- Dòng 1: Chứa số nguyên dương $n \leq 10^5$
- n dòng tiếp theo, dòng thứ i chứa hai số nguyên L_i, R_i ($\forall i: -10^6 \leq L_i < R_i \leq 10^6$)

Kết quả: Ghi ra file văn bản REP.OUT

- Dòng 1: Ghi số điểm được chọn m
- Dòng 2: Ghi các toạ độ (trên trục số) của m điểm được chọn

Các số trên một dòng của Input/Output files được/phải ghi cách nhau ít nhất một dấu cách

Ví dụ:

REP . INP	REP . OUT
3	3
6 10	4 6 9
1 6	
4 9	

ĐOẠN 0

Cho dãy số nguyên $A = (a_1, a_2, \dots, a_n)$ ($1 \leq n \leq 10^5$; $-10^6 \leq a_i \leq 10^6, \forall i: 1 \leq i \leq n$)

Hãy tìm một đoạn dài nhất gồm các phần tử liên tiếp trong dãy $A: (a_L, a_{L+1}, \dots, a_H)$ có tổng bằng 0

Dữ liệu: Vào từ file văn bản SZERO.INP

- Dòng 1: Chứa số n
- Dòng 2: Chứa n số a_1, a_2, \dots, a_n theo đúng thứ tự cách nhau ít nhất một dấu cách

Kết quả: Ghi ra file văn bản SZERO.OUT

Chỉ gồm một dòng ghi hai số L và H cách nhau ít nhất một dấu cách.

Ví dụ:

SZERO.INP	SZERO.OUT
9	2 8
2 7 5 -3 -2 4 -9 -2 1	

Dữ liệu vào luôn được cho hợp lý để tồn tại một đoạn các phần tử liên tiếp trong dãy A có tổng bằng 0.

ĐOẠN DƯƠNG

Cho dãy số nguyên $A = (a_1, a_2, \dots, a_n)$. Hãy tìm một đoạn dài nhất gồm các phần tử liên tiếp trong dãy A : $(a_L, a_{L+1}, \dots, a_H)$ có tổng là số dương.

Dữ liệu: Vào từ file văn bản PS.INP

- Dòng 1: Chứa số nguyên dương $n \leq 10^6$
- Dòng 2: Chứa n số a_1, a_2, \dots, a_n theo đúng thứ tự. ($\forall i: |a_i| \leq 10^6$), có ít nhất một số dương trong dãy.

Kết quả: Ghi ra file văn bản PS.OUT hai chỉ số L và H trên một dòng

Các số trên một dòng của Input/Output files được/phải ghi cách nhau ít nhất một dấu cách

Ví dụ:

PS.INP	PS.OUT
10	3 9
-5 -2 -3 4 -6 7 -8 9 -1 -20	

BỐ TRÍ PHÒNG HỌP

Có n cuộc họp đánh số từ 1 đến n đăng ký làm việc tại một phòng hội thảo. Cuộc họp i cần được bắt đầu ngay sau thời điểm s_i và kết thúc tại thời điểm f_i : $(s_i, f_i]$. Hỏi có thể bố trí phòng hội thảo phục vụ được nhiều nhất bao nhiêu cuộc họp, sao cho khoảng thời gian làm việc của hai cuộc họp bất kỳ là không giao nhau.

Dữ liệu: vào từ file văn bản ACTIVITY.INP

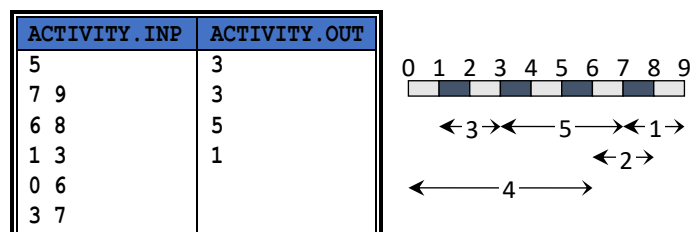
- Dòng đầu tiên chứa số nguyên dương $n \leq 10^6$
- Dòng thứ i trong số n dòng tiếp theo chứa hai số nguyên s_i, f_i . ($0 \leq s_i < f_i \leq 10^6$)

Các số trên một dòng của Input file được ghi cách nhau ít nhất một dấu cách

Kết quả: Ghi ra file ACTIVITY.OUT

- Dòng đầu tiên ghi số k là số các cuộc họp được chấp nhận phục vụ
- k dòng tiếp theo liệt kê số hiệu các cuộc họp được chấp nhận theo thứ tự từ cuộc họp đầu tiên tới cuộc họp cuối cùng, mỗi dòng ghi số hiệu một cuộc họp.

Ví dụ:



MÃ HOÁ BURROWS WHEELER

Mã hóa Burrows–Wheeler là một thuật toán sử dụng trong nén dữ liệu được phát minh ra bởi Michael Burrows and David Wheeler (1994). Định nghĩa từ là một xâu ký tự chỉ gồm chữ cái in hoa ('A'..'Z'). Xét từ W độ dài n . Thuật toán mã hóa có thể mô tả như sau (Ví dụ với từ 'BANANA'):

Bước 1: Viết thêm vào cuối từ ký tự '@', Xét $n + 1$ hoán vị vòng quanh: BANANA@ ANANA@B NANA@BA ANA@BAN NA@BANA A@BANAN @BANANA	Bước 2: Sắp xếp $n + 1$ hoán vị vòng quanh đó theo thứ tự từ điển: @BANANA A@BANAN ANA@BAN ANANA@B BANANA@ NA@BANA NANA@BA	Bước 3: Viết ra các ký tự cuối của các hoán vị vòng quanh theo đúng thứ tự sau khi đã sắp xếp tạo thành từ mã của W ANNB@AA
---	---	--

Yêu cầu: Hãy viết chương trình mã hóa/giải mã Burrows Wheeler.

Dữ liệu: Vào từ file văn bản BWT.INP gồm nhiều dòng, mỗi dòng chứa một từ cần mã hóa/giải mã

Kết quả: Ghi ra file văn bản BWT.OUT có cùng số dòng với file dữ liệu. Trên mỗi dòng ghi kết quả từ mã/giải mã trên dòng tương ứng của file dữ liệu.

Ràng buộc dữ liệu: Dữ liệu luôn được cho đúng đắn. Các từ trong file dữ liệu dài không quá 10^5 ký tự. File dữ liệu có không quá 10 dòng.

Ví dụ:

BWT . INP	BWT . OUT
THEBEATLES	SEEBHLTTE@A
YDRTYEESA@	YESTERDAY
L@LA	ALL
Y@M	MY
SULBRTE@O	TROUBLES
DEMSEE@	SEEMED
OS@	SO
RF@A	FAR
Y@WAA	AWAY