

Bài A. GNUM

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Đất nước xyz tươi đẹp gồm n thành phố, đánh số từ 1 đến n . Một số cặp thành phố được nối với nhau bởi các con đường một chiều, cho phép đi lại theo một chiều nào đó. Theo đó, thành phố u được coi là đến được thành phố v (trực tiếp hoặc gián tiếp) nếu từ u , qua một số con đường và thành phố trung gian ta đến được v . Sắp tới Noel, ông già Noel sẽ cười tuần lộc đi phát quà cho các coder chăm chỉ.

Để chào đón ông, người ta sẽ treo n băng-rôn ở n thành phố, mỗi thành phố một băng-rôn. Mỗi băng-rôn ghi một số nguyên thuộc $[1; n]$ và không có hai băng-rôn nào ghi cùng một số. Ông già Noel không thích sự giảm sút, nếu ông sau khi phát quà ở thành phố u , ông đi sang phát quà ở thành phố v mà số ghi lần lượt ở hai băng-rôn u và v đã giảm đi thì ông sẽ không vui. Để đánh giá mức độ không vui của ông, người ta định nghĩa hệ số phạt của một cách treo băng-rôn như sau:

Gọi a_i là số ghi trên băng-rôn treo ở thành phố i . Mỗi cặp u, v mà u đi đến được v (trực tiếp hoặc qua một số thành phố trung gian) và $a_v < a_u$ gọi là một cặp phạt, giá trị phạt của cặp này là $a_u - a_v$. Hệ số phạt của cách treo là tổng giá trị phạt của tất cả các cặp phạt. Hãy tìm một cách treo các băng-rôn có hệ số phạt nhỏ nhất

Dữ liệu vào

- Dòng đầu tiên chứa n, m là số thành phố và số đường một chiều
- m dòng tiếp theo, mỗi dòng chứa 2 số nguyên u, v mô tả một đường

Kết quả

- Dòng đầu tiên chứa hệ số phạt của cách treo
- Dòng thứ 2 chứa n số nguyên dương mô tả cách treo tìm được, số thứ i là số ghi trên băng-rôn treo ở thành phố i

Ví dụ

stdin	stdout
11 14 4 5 3 1 2 3 3 4 5 4 4 9 10 11 8 7 6 8 7 6 8 11 7 4 1 2 11 10	10 6 7 8 9 10 1 3 2 11 5 4

Hạn chế

- $1 \leq n, m \leq 10^5$
- Có 50% số test với $1 \leq n \leq 1000$

Bài B. GBOX

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Trò chơi lật hộp được mô tả như sau:

Cho một lưới ô vuông có n dòng và m cột, các dòng được đánh số từ 0 đến $n - 1$ từ trên xuống dưới, các cột được đánh số từ 0 đến $m - 1$ từ trái sang phải. Ô ở vị trí dòng i , cột j gọi là ô (i, j) . Tại ô (x, y) có đặt một khối hình hộp chữ nhật kích thước $1 \times 1 \times 3$ (đặt đứng gọn vào ô (x, y)). Bạn cần di chuyển hộp bằng cách lật nó trên lưới theo bất kỳ hướng nào (tất nhiên là lật theo 1 cạnh của hộp đang tiếp xúc với lưới, và không để bất kỳ phần nào của đáy hộp nằm ngoài lưới) để đưa hộp đến đặt đứng gọn ở ô (z, t) . Tuy nhiên, một số ô của lưới đã bị hư hại và không thể di chuyển hộp vào đó (tức là mặt đáy của hộp phải luôn không nằm đè lên bất cứ ô bị hư hại nào). Là một game thủ giỏi, bạn hãy chỉ ra cách di chuyển hộp sao cho số lần lật hộp là ít nhất (nếu có nhiều phương án cùng có số lần lật hộp ít nhất, đưa ra phương án bất kỳ); hoặc thông báo không có cách nào để di chuyển được hộp theo yêu cầu

Dữ liệu vào

- Dòng đầu chứa 6 số nguyên không âm $n\ m\ x\ y\ z\ t$
- n dòng tiếp theo, mỗi dòng chứa m ký tự là 0/1 tương ứng là không bị hoặc bị hư hại. Ký tự thứ j trên dòng thứ i cho biết ô $(i - 1, j - 1)$ có bị hư hại hay không

Dữ liệu đảm bảo $1 \leq n, m \leq 100$. (x, y) , (z, t) là các ô nằm trên lưới và không bị hư hại

Kết quả

- Nếu không có cách nào di chuyển hộp theo yêu cầu, in ra -1. Nếu không:
- Dòng đầu tiên chứa số nguyên dương k : Số lần ít nhất phải lật hộp
- Dòng thứ 2 mô tả phương án lật, gồm k số nguyên thuộc tập $\{0, 1, 2, 3\}$ tương ứng là lật hộp sang phải, xuống dưới, sang trái, lên trên

Ví dụ

stdin	stdout
4 6 0 0 3 5 000000 000001 000000 000000	8 1 0 3 0 1 1 1 0

Bài C. BOX2

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Cho n chiếc hộp được đánh số từ 1 đến n . Hộp thứ i có chiều dài a_i , chiều rộng b_i . Hộp i có thể đặt vào trong hộp j nếu i chưa bị chứa bởi hộp nào khác, j đang không chứa hộp nào khác và $a_i < a_j$, $b_i < b_j$. Cần tìm cách lồng các hộp vào nhau sao cho số hộp không bị lồng vào bất kỳ hộp nào là ít nhất. Nếu có nhiều cách lồng các hộp đều là tốt nhất, in ra cách bất kỳ

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương n
- Dòng thứ i trong số n dòng tiếp theo ghi a_i b_i

Kết quả

- Dòng đầu chứa số nguyên dương k là số hộp còn lại
- k dòng tiếp theo, mỗi dòng mô tả một dãy hộp: Số đầu tiên là số lượng hộp bị lồng vào nhau và theo sau là chỉ số của các hộp bị lồng vào nhau, theo thứ tự từ ngoài vào trong của các hộp

Các dãy hộp có thể in ra theo thứ tự tùy ý

Ví dụ

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

Hạn chế

- $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^9$
- Subtask 1: $n \leq 5000$
- Subtask 2: $a_i = b_i$
- Subtask 3: Ràng buộc gốc

Bài D. FCODE

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Những người bạn thân lâu năm có thể sử dụng một loại ký hiệu khác thường để xác nhận và trò chuyện với nhau, gọi là mật mã huynh đệ. Tuấn và Tú chơi thân với nhau từ nhỏ, và họ đã nghĩ ra mật mã cho riêng mình. Theo đó, mỗi người sẽ chọn ra một xâu gồm n ký tự, mỗi ký tự đều thuộc tập $\{'\{', '[', '(', '}', ']', '\)'\}$. Khi gặp mặt, họ sẽ đem xâu của mình trộn lẫn vào nhau, lần lượt là ký tự đầu tiên trong xâu của Tuấn, rồi đến ký tự đầu tiên trong xâu của Tú, tiếp theo là ký tự thứ hai, và cứ thế. Nếu xâu thu được là dãy ngoặc đúng độ dài $2n$, độ sâu k thì mật mã được xác nhận. Dãy ngoặc đúng được định nghĩa như sau:

- Xâu rỗng là dãy ngoặc đúng bậc 0
- Nếu A là dãy ngoặc đúng bậc a , B là dãy ngoặc đúng bậc b thì: $\{A\}$, (A) , $[A]$ là các dãy ngoặc đúng bậc $a + 1$. AB là dãy ngoặc đúng bậc $\max(a, b)$

Trải qua thời gian, Tuấn không còn nhớ rõ xâu mà ngày đó mình đã chọn. Trong trí nhớ của Tuấn, xâu của anh sẽ trông giống như: $?(??)?\{}$. Còn xâu của Tú thì dĩ nhiên là anh không nhớ gì. Tuấn thắc mắc là có bao nhiêu khả năng có thể có của mật mã huynh đệ, tức là có bao nhiêu dãy ngoặc đúng độ dài $2n$ độ sâu k sao cho các vị trí không phải dấu "?" trên xâu của Tuấn sau khi trộn lại đều trùng khớp trên dãy ngoặc đó

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương: n k
- Dòng thứ hai chứa một xâu có độ dài n , gồm các ký tự thuộc tập $\{'\{', '[', '(', '}', ']', '\)'\}$

Kết quả

- Ghi số cách khôi phục tìm được, sau khi chia lấy dư cho $10^9 + 7$

Ví dụ

stdin	stdout
8 3 ?(??)?{}	4617
2 1 (?	3

Hạn chế

- $1 \leq k \leq n \leq 1000$
- Subtask 1: $n \leq 50$
- Subtask 2: $k \leq 5$
- Subtask 3: Ràng buộc gốc

Bài E. INVMX

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

Cho hai số nguyên dương n, k và một dãy số nguyên dương $a = a_1, a_2, \dots, a_n$ có các phần tử không quá k . Một số vị trí trên dãy a đã bị mất, được đánh dấu bởi số 0. Bạn cần khôi phục lại a sao cho a có nhiều nghịch thế nhất. Cụ thể hơn, cần tìm cách thay thế các số 0 trong dãy a bởi các số nguyên dương không quá k sao cho số cặp (i, j) mà $i < j$ và $a_i > a_j$ là lớn nhất.

Dữ liệu vào

- Dòng đầu chứa n, k
- Dòng tiếp theo chứa dãy a

Kết quả

- Ghi số nghịch thế lớn nhất có thể

Ví dụ

stdin	stdout
6 9 0 8 4 3 0 0	15
10 9 5 2 9 0 7 4 8 7 0 0	28
10 9 7 4 0 0 8 5 0 0 3 1	36

Hạn chế

- $n \leq 2 \times 10^5, 1 \leq k \leq 100$

- Có 50% số test với $n \leq 1000$