

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. TGAMEZ

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

Cây là một đồ thị vô hướng liên thông không có chu trình. Cho một cây có n đỉnh, các đỉnh được đánh số từ 1 đến n với gốc là đỉnh 1. Nút thứ i của cây được đặt a_i đồng xu

Xét một trò chơi thú vị trên cây này như sau:

- Có hai người chơi. Hai người chơi này sẽ luân phiên thực hiện lượt chơi của mình
- Mỗi lượt chơi, người chơi chọn ra một nút không phải là nút gốc của cây và chuyển một số đồng xu ở đó lên nút cha của nó (ít nhất là một và nhiều nhất là toàn bộ)
- Ai không thực hiện được lượt chơi nữa thì thua cuộc

Hãy tính xem người đi trước có bao nhiêu cách thực hiện nước đi đầu tiên sao cho chắc chắn dành được chiến thắng. Hai cách đi được coi là khác nhau nếu nút được chọn khác nhau hoặc số lượng đồng xu được chuyển đi khác nhau

Dữ liệu vào

- Dòng đầu chứa số nguyên dương: n
- $n - 1$ dòng tiếp theo, dòng thứ i chứa hai số nguyên không âm là đỉnh cha của đỉnh $i + 1$ và a_{i+1}

a_1 không ảnh hưởng đến trò chơi nên không cần nhập vào

Kết quả

Ghi số cách tìm được

Ví dụ

stdin	stdout
4 1 2 1 4 1 5	1

Hạn chế

- $2 \leq n \leq 10^5$, $0 \leq a_i \leq 10^9$
- Subtask 1: Độ sâu của cây (số cạnh nhiều nhất trên đường đi từ gốc đến nút lá) bằng 1
- Subtask 2: Độ sâu của cây (số cạnh nhiều nhất trên đường đi từ gốc đến nút lá) bằng $n - 1$
- Subtask 3: Ràng buộc gốc

Bài C. CCIRCLE

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

Cho s và t là hai chuỗi nhị phân có cùng độ dài và độ dài không quá 100. Hãy sử dụng không quá 10^5 phép biến đổi để biến s thành t , mỗi biến đổi có dạng như sau:

- Đánh số các ký tự trên s bắt đầu từ 0
- Chọn vị trí i , $0 \leq i < |s|$ sao cho $s_i = s_{(i+1) \% |s|}$
- Đảo ngược (0 thành 1 và 1 thành 0) s_i và $s_{(i+1) \% |s|}$

Dữ liệu vào

- Dòng đầu chứa chuỗi s
- Dòng thứ hai chứa chuỗi t

Kết quả

- Dòng đầu ghi số phép biến đổi cần dùng: m
- Dòng tiếp theo ghi m số là vị trí i trong bước chọn vị trí của phép biến đổi

Dữ liệu đảm bảo có nghiệm. Lưu ý bạn không cần cực tiểu m . Nếu có nhiều dãy biến đổi không quá 10^5 phép, chỉ cần đưa ra một dãy bất kỳ trong số đó

Ví dụ

stdin	stdout
000	1
011	1
110	2
011	0
	1

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. LNET

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

Hệ thống mạng trên hành tinh XYZ thỏa mãn tính chất sau: Giữa hai đỉnh x, y bất kỳ, tồn tại và duy nhất một đường đi đơn giữa chúng và được ký hiệu là $\text{Path}(x, y)$. Nói cách khác, hệ thống mạng có dạng hình cây. Có một số cặp nút mạng đang truyền thông tin cho nhau, gọi là các kết nối. Với a là một kết nối, ký hiệu s_a và t_a lần lượt là đỉnh gửi và đỉnh nhận ($s_a \neq t_a$).

Ta nói kết nối a và kết nối b là xung đột nhau nếu chúng dùng chung ít nhất một nút mạng, tức là $\text{Path}(s_a, t_a)$ và $\text{Path}(s_b, t_b)$ có đỉnh chung. Lúc này, cần phải dùng hai cách mã hóa khác nhau để mã hóa cho đường truyền a và b . Câu hỏi đặt ra là: Cần dùng ít nhất bao nhiêu cách mã hóa khác nhau để gán cho mỗi kết nối một cách mã hóa, đảm bảo hai kết nối xung đột nhau thì dùng hai cách mã hóa khác nhau.

Các kết nối trên mạng có tính trực tuyến. Ban đầu chưa có kết nối nào, sau đó có thể có thêm các kết nối hoặc một số kết nối mất đi. Sau mỗi lần biến đổi như vậy, hãy tính toán và đưa ra số cách mã hóa ít nhất cần dùng.

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương: n m là số đỉnh của cây và số thay đổi của mạng
- n dòng tiếp theo mỗi dòng ghi một cạnh của cây: u v
- m dòng tiếp theo mỗi dòng ghi một biến đổi của mạng: k s t với s là đỉnh gửi, t là đỉnh nhận, $k = 0/1$ tương ứng là có thêm hoặc mất đi một kết nối từ s đến t

Dữ liệu đảm bảo có ít nhất một kết nối từ s đến t khi $k = 1$, và nếu có nhiều kết nối từ s đến t thì mỗi lần chỉ mất đi một trong số đó. Các đỉnh của cây được đánh số từ 1.

Kết quả

- Ghi m dòng là kết quả tính được sau mỗi biến đổi

Ví dụ

stdin	stdout
7 5	1
1 2	2
1 3	3
2 4	4
2 5	3
3 6	
3 7	
0 2 7	
0 3 1	
0 1 4	
0 4 6	
1 2 7	

Hạn chế

- $1 \leq n, m \leq 10^5$
- Subtask 1: $1 \leq n, m \leq 1000$
- Subtask 2: $1 \leq m \leq 1000$
- Subtask 3: $1 \leq n \leq 1000$
- Subtask 4: Ràng buộc gốc