

2023 summer contest #29, university training, round A

Statement
is not
available
on
English
language

D. Truy vấn trọng số cực đại

0.25 seconds, 128 megabytes

Cho một cây gồm n đỉnh, các đỉnh đánh số từ 1 tới n . Mỗi cạnh trên cây có một trọng số nguyên dương.

Với hai đỉnh phân biệt x và y , ta định nghĩa khoảng cách giữa x và y là **trọng số lớn nhất** của một cạnh nằm trên **đường đi đơn duy nhất** giữa x và y .

Bạn cần trả lời q truy vấn. Mỗi truy vấn cho hai số nguyên dương l và r ($l \leq r$), bạn cần đếm số cặp đỉnh (x, y) ($x < y$) cho khoảng cách giữa x và y nằm trong đoạn $[l, r]$.

Input

Dòng đầu tiên chứa hai số nguyên n và q ($1 \leq n, q \leq 10^5$) — số đỉnh của cây và số truy vấn.

$n - 1$ dòng tiếp theo, mỗi dòng chứa ba số nguyên u, v và w ($1 \leq u, v \leq n, 1 \leq w \leq 10^9, \frac{u+v}{2} > \sqrt{uv}$), thể hiện một cạnh nói giữa u và v có trọng số w .

q dòng cuối cùng, mỗi dòng chứa hai số nguyên l và r ($1 \leq l \leq r \leq 10^9$) thể hiện một truy vấn.

Output

In ra q số nguyên là kết quả của q truy vấn. Các số được viết trên một dòng, ngăn cách bởi dấu cách.

Scoring

- Subtask 1 (30 điểm): $1 \leq n, q \leq 10^3$.
- Subtask 2 (70 điểm): $1 \leq n, q \leq 10^5$.

input
5 5
1 2 3
1 4 2
2 5 6
3 4 1
1 1
1 2
2 3
2 5
1 6
output
1 3 5 5 10

Statement
is not
available
on

English
language

E. Ôn tập Hóa học

0.75 seconds, 256 megabytes

Do dịch COVID-19 kéo dài, các kỳ thi học sinh giỏi quốc tế có nguy cơ bị hủy làm cho phong trào học tin trong cả nước có phần trùng xuống. Trong khi đó, nhu cầu ôn tập Hóa học lại đang tăng cao đột biến trước sự bất định của kỳ thi THPT QG năm nay. Đứng trước nguy cơ số buổi dạy đội tuyển tin giảm sút, GS PVH quyết định đầu tư mở lớp ôn thi cấp tốc Hóa học.

Để thực hiện kế hoạch của mình, GS PVH mời n chuyên gia hóa học về giảng dạy. Theo đánh giá của GS, chuyên gia i có trình độ là v_i . Có ba chủ đề lớn của Hóa học THPT: *Phản ứng oxi hóa-khử*, *Hóa học vô cơ* và *Hóa học hữu cơ*. GS sẽ chia n chuyên gia này làm ba nhóm để dạy ba chủ đề này. **Mỗi chuyên gia thuộc về đúng một nhóm, và mỗi nhóm phải có ít nhất một chuyên gia.**

Nghiên cứu cho thấy, chất lượng của một nhóm chuyên gia bằng **tổng xor** trình độ của các chuyên gia trong nhóm. GS PVH muốn chia n chuyên gia này thành ba nhóm sao cho **tổng chất lượng của ba nhóm là lớn nhất**.

Nhắc lại, phép toán \oplus (xor) đối với bit được định nghĩa như sau:

- $0 \oplus 0 = 0$,
- $1 \oplus 0 = 1$,
- $0 \oplus 1 = 1$,
- $1 \oplus 1 = 0$.

Phép toán \oplus đối với các số có nhiều hơn một bit được thực hiện theo từng bit. Ví dụ:

- $2 \oplus 3 = 1 (10_2 \oplus 11_2 = 01_2)$,
- $2 \oplus 5 = 7 (010_2 \oplus 101_2 = 111_2)$,
- $5 \oplus 5 = 0 (101_2 \oplus 101_2 = 000_2)$.

Input

Dòng đầu tiên chứa số nguyên t ($1 \leq t \leq 4$) — số thứ tự của subtask chứa test này.

Các dòng tiếp theo mô tả các bộ dữ liệu. Mỗi bộ dữ liệu được mô tả bởi hai dòng: Dòng đầu tiên chứa số nguyên n ($3 \leq n \leq 200$) — số chuyên gia được GS PVH mời về dạy. Dòng thứ hai chứa n số nguyên v_1, v_2, \dots, v_n ($0 \leq v_i \leq 600$) — trình độ của các chuyên gia.

Dòng cuối cùng của file input chứa số 0 và bạn không phải xử lý dòng này.

Mỗi file input có không quá 7 bộ dữ liệu. Tổng giá trị n trong tất cả các bộ dữ liệu của một file input không quá 250.

Output

Với mỗi bộ dữ liệu, in ra trên một dòng xâu ký tự độ dài n thể hiện nhóm mà các chuyên gia được phân vào. Xâu chỉ chứa các ký tự 'P', 'V' và 'H' trong đó:

- ký tự thứ i là 'P' khi và chỉ khi chuyên gia thứ i được phân công dạy về *phản ứng oxi hóa-khử*,
- ký tự thứ j là 'V' khi và chỉ khi chuyên gia thứ j được phân công dạy về *Hóa học vô cơ*,

- ký tự thứ k là 'H' khi và chỉ khi chuyên gia thứ k được phân công dạy về *Hóa học hữu cơ*.

Lưu ý, mỗi nhóm cần có ít nhất một chuyên gia và tổng chất lượng của ba nhóm phải lớn nhất có thể. Nếu có nhiều phương án tối ưu, bạn có thể in ra phương án bất kỳ.

Scoring

- Subtask 1 (26 điểm): $n \leq 15$.
- Subtask 2 (18 điểm): $0 \leq v_i \leq 1$.
- Subtask 3 (18 điểm): $v_1 = v_2 = \dots = v_n$.
- Subtask 4 (38 điểm): Không có ràng buộc gì thêm.

Ngoài ra, với mỗi test (mỗi file input), bạn sẽ được 15% số điểm của test đó với mỗi bộ dữ liệu có kết quả chính xác trước bộ dữ liệu đầu tiên bị sai. Ví dụ, nếu trong một file input, bạn đưa ra kết quả đúng cho 3 bộ dữ liệu đầu tiên và sai ở bộ dữ liệu thứ tư, bạn được 45% số điểm của file input này.

input
1 3 2 3 3 4 7 3 5 2 0
output
HPV PHVV

Statement
is not
available
on
English
language

F. Cây tương đồng

1.5 seconds, 256 megabytes

Cho hai cây gồm n đỉnh, các đỉnh của mỗi cây được đánh số từ 1 tới n . Nhắc lại, cây là một **đồ thị vô hướng liên thông** và **không có chu trình**. Có thể chứng minh rằng một cây gồm n đỉnh luôn có $n - 1$ cạnh.

Xét cạnh e_1 nào đó nối hai đỉnh u_1 và v_1 ở cây thứ nhất, và một cạnh e_2 nào đó nối hai đỉnh u_2 và v_2 ở cây thứ hai. Ta định nghĩa *độ tương đồng* giữa hai cạnh e_1 và e_2 , ký hiệu là $S(e_1, e_2)$ hay $S(u_1, v_1, u_2, v_2)$ như sau:

- Giả sử ta xóa cạnh e_1 nối u_1 và v_1 khỏi cây thứ nhất, khi đó cây bị tách thành hai thành phần liên thông. Gọi tập hợp các nhãn của các đỉnh thuộc hai thành phần liên thông này lần lượt là A_1 và A_2 .
- Giả sử ta xóa cạnh e_2 nối u_2 và v_2 khỏi cây thứ hai, khi đó cây bị tách thành hai thành phần liên thông. Gọi tập hợp các nhãn của các đỉnh thuộc hai thành phần liên thông này lần lượt là B_1 và B_2 .
- Độ tương đồng* giữa hai cạnh e_1 và e_2 được xác định bởi $S(e_1, e_2) = S(u_1, v_1, u_2, v_2) = \max |A_i \cap B_j|$ với $\{i, j\} \subset \{1, 2\}$. Nói cách khác, độ tương đồng $S(e_1, e_2)$ chính là **số nhãn chung lớn nhất** của một thành phần liên thông bị tách ra ở cây thứ nhất và một thành phần liên thông bị tách ra ở cây thứ hai.

Để hiểu rõ hơn, xin vui lòng xem trong phần ghi chú giải thích test ví dụ ở cuối bài.

Problems - Codeforces

Dễ thấy rằng với mọi cặp cạnh (d, e) bất kì trong đó d là một cạnh của cây thứ nhất và e là một cạnh của cây thứ hai, ta luôn có $0 \leq S(d, e) \leq n$. Với mỗi số nguyên x thỏa mãn $0 \leq x \leq n$, bạn cần trả lời câu hỏi: Có bao nhiêu cặp cạnh (d, e) thỏa mãn $S(d, e) = x$, trong tổng cộng $(n - 1)^2$ cặp cạnh (d, e) khác nhau?

Input

- Dòng đầu tiên chứa số nguyên n ($2 \leq n \leq 4000$) — số đỉnh của mỗi cây.
- $n - 1$ dòng tiếp theo, mỗi dòng chứa hai số nguyên u và v ($1 \leq u, v \leq n, \frac{u+v}{2} > \sqrt{uv}$) thể hiện một cạnh của cây thứ nhất.
- $n - 1$ dòng cuối cùng, mỗi dòng chứa hai số nguyên u và v ($1 \leq u, v \leq n, \frac{u+v}{2} > \sqrt{uv}$) thể hiện một cạnh của cây thứ hai.

Dữ liệu vào đảm bảo cả hai cây đều không có chu trình.

Output

In ra $n + 1$ số nguyên c_0, c_1, \dots, c_n , trong đó c_k bằng số cặp cạnh (d, e) trong đó cạnh d thuộc cây thứ nhất, e thuộc cây thứ hai và $S(d, e) = k$. Các số được viết trên một dòng, ngăn với nhau bởi dấu cách.

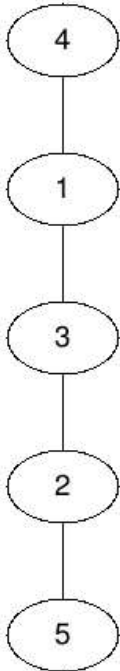
input
5 1 3 1 4 2 3 2 5 1 2 2 5 3 5 4 5
output
0 0 8 7 1 0

input
2 1 2 1 2
output
0 1 0

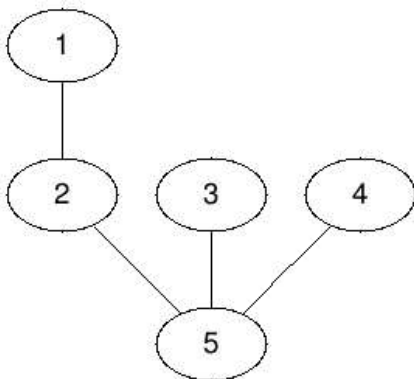
input
7 6 3 1 4 4 5 4 7 2 6 6 4 6 3 1 4 4 5 4 7 2 6 6 4
output
0 0 0 6 5 20 5 0

Hình vẽ dưới đây mô tả hai cây trong test ví dụ đầu tiên:

Cây thứ nhất:



Cây thứ hai:



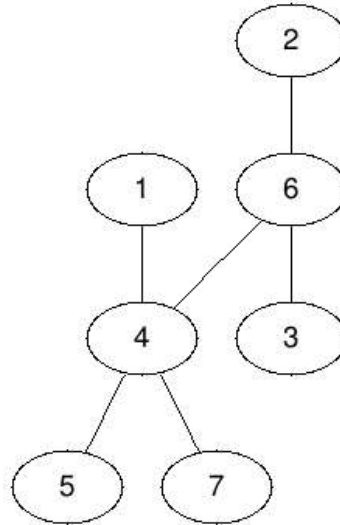
Xét cạnh $(1, 3)$ ở cây thứ nhất và cạnh $(1, 2)$ ở cây thứ hai, độ tương đồng giữa hai cạnh này được xác định như sau:

- Nếu xóa cạnh $(1, 3)$ khỏi cây thứ nhất, ta được hai thành phần liên thông $A_1 = \{1, 4\}$ và $A_2 = \{2, 3, 5\}$.
- Nếu xóa cạnh $(1, 2)$ khỏi cây thứ hai, ta được hai thành phần liên thông $B_1 = \{1\}$ và $B_2 = \{2, 3, 4, 5\}$.
- Giờ ta cần chọn một cặp liên thông $(A_1 \text{ hoặc } A_2)$ với $(B_1 \text{ hoặc } B_2)$ sao cho hai tập hợp được chọn có nhiều nhân chung nhất. Cách chọn tối ưu là A_2 và B_2 , khi đó $|A_2 \cap B_2| = |\{2, 3, 5\}| = 3$. Do đó độ tương đồng của hai cạnh đang xét là 3.

Tương tự, để xác định độ tương đồng giữa cạnh $(2, 3)$ ở cây thứ nhất và cạnh $(4, 5)$ ở cây thứ hai, ta có:

- Hai thành phần liên thông bị tách ra khi xóa cạnh $(2, 3)$ khỏi cây thứ nhất: $A_1 = \{2, 5\}$ và $A_2 = \{1, 3, 4\}$.
- Hai thành phần liên thông bị tách ra khi xóa cạnh $(4, 5)$ khỏi cây thứ hai: $B_1 = \{4\}$ và $B_2 = \{1, 2, 3, 5\}$.
- Độ tương đồng của hai cạnh này là 2, do $\max |A_i \cap B_j| = |A_1 \cap B_2| = 2$.

Trong ví dụ thứ ba, hai cây đều giống nhau và như hình dưới đây:



Statement
is not
available
on
English
language

G. Kẹo gừng

0.3 seconds, 30 megabytes

Giang là một trong những học sinh xuất sắc nhất của trường THPT chuyên Bắp Rang. Dù đã thể hiện ấn tượng trong vòng thi ICPC vừa qua và dành một suất vào vòng sau, niềm vui của Giang không được trọn vẹn khi cô bạn thân nhất của anh không thể đạt được kết quả tương tự. Để an ủi cô bạn đáng thương, Giang muốn mua một vài chiếc kẹo gừng — món ăn ưa thích của cô gái.

Hệ thống giao thông tại thành phố Bắp Rang có n giao lộ được nối với nhau bởi m con đường hai chiều. Các giao lộ được đánh số từ 1 tới n và các con đường được đánh số từ 1 tới m . Trên con đường thứ i nối hai giao lộ u_i và v_i , có một cửa hàng bán c_i cái kẹo gừng. Không có hai con đường nào có cùng số kẹo được bán. Giang hẹn gặp bạn thân của mình ở một giao lộ nào đó, đi qua vài con đường nhưng **không đi qua một con đường nào hai lần**, mua **toàn bộ số kẹo** được bán ở những con đường sẽ đi qua, và **trở lại giao lộ được chọn làm điểm hẹn**.

Sử dụng vốn kiến thức VL eo hẹp của mình, Giang tính toán lượng calories Giang tiêu thụ cho quãng đường di chuyển như sau: Gọi L là **số kẹo gừng lớn nhất** anh ta mua trên một con đường, và K là số con đường anh ta đã đi qua. Lượng calories Giang tiêu thụ sẽ là $L^2 + \alpha K$, với α là một hằng số anh ta đã biết từ trước.

Hãy giúp Giang xác định điểm hẹn gặp bạn thân, các con đường Giang đi qua và mua kẹo sao cho lượng năng lượng tiêu thụ là nhỏ nhất.

Input

- Dòng đầu tiên chứa ba số nguyên n , m và α ($1 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$, $1 \leq \alpha \leq 20$), lần lượt là số giao lộ, số con đường hai chiều và hằng số α Giang sử dụng để xác định lượng calories tiêu thụ.
- Trong m dòng tiếp theo, dòng thứ i chứa ba số nguyên u_i , v_i và c_i ($1 \leq u_i, v_i \leq n$, $10^6 \leq c_i \leq 10^9$), cho biết con đường thứ i nối hai giao lộ u_i và v_i và cửa hàng kẹo gừng trên con đường này bán c_i chiếc.

Dữ liệu vào đảm bảo các giá trị c_i ở m dòng trên đôi một phân biệt.

Output

Dòng đầu tiên chứa một số nguyên là số calories nhỏ nhất Giang tiêu thụ trên quãng đường di chuyển, hoặc "Poor girl" nếu Giang không thể tìm ra hành trình thỏa mãn các điều kiện trên.

Nếu tồn tại một hành trình thỏa mãn:

- Dòng thứ hai chứa hai số nguyên lần lượt là giao lộ Giang hẹn bạn thân và số con đường Giang sẽ đi.
- Dòng thứ ba chứa các số nguyên là chỉ số của các con đường mà Giang sẽ đi qua.

Nếu có nhiều phương án tối ưu, bạn được in ra phương án bất kì.

input
7 7 10 1 2 1000000 2 3 2000000 3 4 3000000 4 5 4000000 5 6 5000000 6 7 6000000 7 1 7000000
output
49000000000070 1 7 1 2 3 4 5 6 7

input
6 6 7 1 3 1000000 3 5 3000000 5 1 5000000 2 4 2000000 4 6 4000000 6 2 6000000
output
25000000000021 1 3 1 2 3

input
5 4 1 1 2 22081999 1 3 28021999 2 4 19992208 2 5 19992802
output
Poor girl

Statement
is not
available
on
English
language

H. Lại là truy vấn trên cây

1.75 seconds, 256 megabytes

Cho một cây gồm n đỉnh. Các đỉnh được đánh số từ 1 tới n . Đỉnh 1 là gốc của cây.

Problems - Codeforces

Trên mỗi đỉnh lưu một số nguyên. Ban đầu giá trị ở tất cả các đỉnh là 0. Bạn cần thực hiện q truy vấn thuộc một trong ba dạng sau:

- add u* δ : Ta thay đổi giá trị các đỉnh thuộc cây con gốc u theo quy tắc:
 - Giá trị của đỉnh u được tăng lên δ .
 - Giá trị các đỉnh là con trực tiếp của u bị giảm đi δ .
 - Giá trị các đỉnh là con trực tiếp của con trực tiếp của u được tăng lên δ .
 - Giá trị các đỉnh là con trực tiếp của con trực tiếp của con trực tiếp của u bị giảm đi δ ...
- get u*: Tính giá trị của đỉnh u .
- pos u k*: Tính giá trị lớn thứ k trong tập giá trị các đỉnh thuộc cây con gốc u .

Yêu cầu: In ra kết quả các truy vấn loại 2 và 3. Với truy vấn loại 3, nếu cây con gốc u có ít hơn k đỉnh, kết quả của truy vấn này là 0.

Input

- Dòng đầu tiên chứa số nguyên t ($1 \leq t \leq 4$) — số thứ tự của subtask chứa test này.
- Dòng thứ hai chứa số nguyên n ($2 \leq n \leq 3 \cdot 10^5$) — số đỉnh của cây.
- Dòng thứ ba chứa $n - 1$ số nguyên p_2, p_3, \dots, p_n ($1 \leq p_i \leq i - 1$), trong đó p_i là cha trực tiếp của đỉnh i .
- Dòng thứ tư chứa số nguyên q ($1 \leq q \leq 3 \cdot 10^5$) — số truy vấn cần thực hiện.
- q dòng cuối cùng, mỗi dòng mô tả một truy vấn theo một trong ba định dạng *add u* δ , *get u* hoặc *pos u k*. Các tham số trên thỏa mãn $1 \leq u \leq n$, $0 \leq |\delta| \leq 7000$ và $1 \leq k \leq 15$.

Output

In ra kết quả của các truy vấn loại 2 và loại 3, theo thứ tự chúng xuất hiện trong input. Các số được viết trên một dòng, ngăn cách với nhau bởi dấu cách.

Scoring

- Subtask 1 (29 điểm): $1 \leq n, q \leq 8000$.
- Subtask 2 (17 điểm): Không có truy vấn loại 1 (*add*).
- Subtask 3 (23 điểm): Trong mọi truy vấn loại 3 (*pos*), $k = 1$.
- Subtask 4 (31 điểm): Không có ràng buộc gì thêm.

input
1 7 1 2 2 4 1 6 18 add 1 1 get 1 get 2 get 3 get 4 get 5 get 6 get 7 add 2 5 add 7 -3 pos 1 1 pos 1 2 pos 1 3 pos 1 4 pos 1 5 pos 1 6 pos 1 7 pos 1 8

output
1 -1 1 1 -1 -1 1 4 4 1 -1 -2 -4 -4 0

input
1 2 1 1 get 1
output
0

input
1 3 1 1 5 add 2 7000 add 3 -7000 pos 1 1 pos 1 2 pos 1 3
output
7000 0 -7000

Statement
is not
available
on
English
language

I. Tô màu

0.25 seconds, 64 megabytes

Cho một bảng ô vuông kích thước $n \cdot n$, trong đó n là một số chẵn. Ban đầu các ô của bảng được tô bởi một trong hai màu trắng hoặc đen.

Bạn được phép tô màu lại bảng này bằng cách sau: Ở mỗi bước, bạn chọn một hàng bất kì của bảng, sau đó thực hiện **đúng một trong hai** thao tác dưới đây:

- Tô toàn bộ các ô đen trên hàng bằng màu trắng.
- Tô toàn bộ các ô trắng trên hàng bằng màu đen.

Bạn muốn bảng của bạn thật đẹp, cụ thể là trên mỗi cột, **không tồn tại** $\frac{n}{2} + 1$ ô vuông liên tiếp nào có cùng màu. Đếm số bước tối thiểu cần để thực hiện điều này.

Input

Dòng đầu tiên chứa một số nguyên chẵn n ($2 \leq n \leq 50$) là kích thước của bảng. n dòng tiếp theo, mỗi dòng gồm n ký tự 'B' hoặc 'W' mô tả trạng thái của bảng ban đầu. Ký tự 'B' thể hiện một ô màu đen, ký tự 'W' thể hiện một ô màu trắng.

Output

In ra một số nguyên duy nhất là số bước tô màu tối thiểu để có được một bảng đẹp. Nếu không tồn tại bảng đẹp nào, in ra số -1.

input
2 WB BB

output
1

input
2 WB WW
output
1

input
2 WB WB
output
2

Statement
is not
available
on
English
language

J. Hướng dẫn viên du lịch

1.5 seconds, 768 megabytes

Đã chán với việc suốt ngày ngồi nhà chỉ có code và code, GS. PVH quyết định đổi nghề, đi làm hướng dẫn viên du lịch cho các vị khách nước ngoài.

Trong một ngày đầu xuân, GS. PVH nhận làm hướng dẫn viên du lịch cho n vị khách đến từ nhiều quốc gia khác nhau. GS. PVH đánh số các vị khách theo thứ tự từ 1 tới n và yêu cầu họ xếp hàng theo đúng thứ tự này. Sau đó, để tiện cho việc tổ chức các hoạt động vui chơi giải trí, GS. PVH chia n hành khách này thành nhiều nhóm, sao cho mỗi nhóm chứa một dãy liên tiếp các hành khách, và mỗi hành khách thuộc vào chính xác một nhóm.

Ngoài việc giúp hành khách có chuyến tham quan thắng cảnh, thường ngoạn thiên nhiên lý thú, GS. PVH còn muốn tạo những tình bạn đẹp giữa những người khách thuộc nhiều nước khác nhau. Nếu trong một nhóm, có trên một nửa số khách thuộc về cùng một quốc gia, họ sẽ sử dụng ngôn ngữ riêng của họ, và rất hạn chế nói chuyện với những khách quốc tế. Bởi vậy, GS. PVH muốn chia nhóm sao cho mỗi nhóm có ít nhất hai khách, và không có quốc gia nào chiếm quá một nửa số khách trong một nhóm.

Bạn hãy giúp GS. PVH đếm số cách chia nhóm thỏa mãn.

Input

Gồm một dòng duy nhất chứa xâu ký tự s gồm không quá 2500000 ký tự, thể hiện quốc tịch của các hành khách. Hành khách thứ i và hành khách thứ j mang cùng quốc tịch khi và chỉ khi $s_i = s_j$. Tất cả các ký tự trong xâu đều là các chữ cái in thường.

Output

Gồm một số nguyên duy nhất là số cách chia nhóm. Do kết quả có thể rất lớn, bạn chỉ cần in ra kết quả theo modulo $10^9 + 22071997$.

Scoring

- Subtask 1 (15 điểm): Xâu s có không quá 100 ký tự.

- Subtask 2 (21 điểm): Xâu s có không quá 9000 kí tự.
- Subtask 3 (19 điểm): Xâu s chỉ chứa chữ cái a và chữ cái b .
- Subtask 4 (21 điểm): Xâu s có không quá 400000 kí tự.
- Subtask 5 (24 điểm): Không có ràng buộc gì thêm.

input
hihi
output
2

input
trasua
output
5

input
emmamckeon
output
20

Trong ví dụ thứ nhất, có hai cách chia nhóm là $\{1 \rightarrow 4\}$ và $\{1 \rightarrow 2, 3 \rightarrow 4\}$.

Trong ví dụ thứ hai, có năm cách chia nhóm là $\{1 \rightarrow 2, 3 \rightarrow 4, 5 \rightarrow 6\}$, $\{1 \rightarrow 2, 3 \rightarrow 6\}$, $\{1 \rightarrow 3, 4 \rightarrow 6\}$, $\{1 \rightarrow 4, 5 \rightarrow 6\}$ và $\{1 \rightarrow 6\}$.

Statement
is not
available
on
English
language

K. Cắm hoa

0.75 seconds, 256 megabytes

Sau khi trở về từ thành phố Đà Lạt mộng mơ, GS. PVH đem những bông hoa xinh tươi ra cắm quanh phòng, tạo nên sự tươi thắm, rực rỡ cho phòng học của mình. GS kiếm được n bông hoa, được đánh số từ 1 tới n . Tiếc thay, GS chỉ có l lọ hoa và mỗi lọ hoa chỉ cắm được một bông. Vì vậy GS có thể phải chia tay với một số bông hoa của mình. Hơn nữa, dựa theo phong thủy và cách bài trí, phối cảnh trong phòng, GS nhận ra rằng bông hoa thứ i chỉ có thể được cắm ở một trong hai lọ a_i hoặc b_i .

Để quyết định xem bông hoa nào sẽ được cắm vào lọ, GS. PVH sẽ xét từng bông hoa theo thứ tự từ 1 tới n . Với bông hoa thứ i , GS áp dụng lần lượt các quy tắc sau đây:

- Nếu lọ hoa a_i đang rỗng, GS sẽ cắm bông hoa thứ i trong lọ a_i . Ngược lại,
- Nếu lọ hoa b_i đang rỗng, GS sẽ cắm bông hoa thứ i trong lọ b_i . Ngược lại,
- Nếu lọ hoa a_i đang có bông hoa j nào đó, thử chuyển bông hoa j này sang lọ hoa còn lại (lưu ý mỗi bông hoa chỉ có thể cắm ở một trong hai lọ hoa). Nếu lọ hoa còn lại này đang chứa bông hoa k nào đó, lại thử chuyển bông hoa này tới lọ hoa còn lại có thể chứa bông hoa k , ... Nếu quá trình này dừng lại khi gặp một lọ hoa đang trống, cắm bông hoa thứ i trong lọ a_i . Ngược lại,

Problems - Codeforces

- Nếu lọ hoa b_i đang có bông hoa j nào đó, thử chuyển bông hoa j này sang lọ hoa còn lại (lưu ý mỗi bông hoa chỉ có thể cắm ở một trong hai lọ hoa). Nếu lọ hoa còn lại này đang chứa bông hoa k nào đó, lại thử chuyển bông hoa này tới lọ hoa còn lại có thể chứa bông hoa k , ... Nếu quá trình này dừng lại khi gặp một lọ hoa đang trống, cắm bông hoa thứ i trong lọ b_i . Ngược lại,
- Đành lòng tạm biệt bông hoa xinh đẹp này.

Cho biết số bông hoa, số lọ hoa GS có và những lọ hoa mà mỗi bông hoa có thể được cắm vào, hãy xác định xem những bông hoa nào được cắm vào lọ, những bông hoa nào không.

Input

Dòng đầu tiên chứa hai số nguyên n và l ($1 \leq n, l \leq 3 \cdot 10^5$), lần lượt là số bông hoa và số lọ hoa GS. PVH có.

n dòng tiếp theo, dòng thứ i chứa hai số nguyên a_i và b_i ($1 \leq a_i, b_i \leq l, \sqrt{\frac{a_i^2 + b_i^2}{2}} > \frac{2}{\frac{1}{a_i} + \frac{1}{b_i}}$) cho biết bông hoa thứ i có thể được cắm ở một trong hai lọ a_i và b_i .

Output

In ra n dòng, dòng thứ i chứa từ **YES** nếu bông hoa thứ i được cắm vào một lọ nào đó, **NO** nếu ngược lại.

Scoring

- Subtask 1 (50 điểm): $n, l \leq 2000$
- Subtask 2 (50 điểm): $n, l \leq 300000$

input
5 3 1 2 1 3 1 2 1 3 1 2
output
Yes Yes Yes No No

input
9 10 1 2 3 4 5 6 7 8 9 10 2 3 1 5 8 2 7 9
output
Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes

Trong ví dụ thứ nhất:

- Bông hoa số 1 được cắm vào lọ hoa số 1 theo luật thứ nhất.

- Bông hoa số 2 được cắm vào lọ hoa số 3 theo luật thứ hai.
- Bông hoa số 3 được cắm vào lọ hoa số 2 theo luật thứ hai.
- Bông hoa số 4 và bông hoa số 5 không thể được cắm vào lọ, do lúc này cả 3 lọ hoa đều không còn trống.

Trong ví dụ thứ hai:

- Áp dụng luật thứ nhất, các bông hoa số từ 1 tới 6 lần lượt được cắm vào các lọ hoa số 1, 3, 5, 7, 9, 2.
- Với bông hoa số 7, ta áp dụng luật thứ ba:
 - Chuyển bông hoa đang cắm ở lọ hoa số 1 sang lọ hoa số 2.
 - Chuyển bông hoa đang cắm ở lọ hoa số 2 sang lọ hoa số 3.
 - Chuyển bông hoa đang cắm ở lọ hoa số 3 sang lọ hoa số 4.
 - Do lọ hoa số 4 đang trống, luật này thành công và bông hoa số 7 được cắm vào lọ số 1.
- Bông hoa số 8 được cắm vào lọ hoa số 8 theo luật thứ nhất.
- Với bông hoa số 9, ta áp dụng luật thứ ba:
 - Chuyển bông hoa đang cắm ở lọ hoa số 7 sang lọ hoa số 8.
 - Chuyển bông hoa đang cắm ở lọ hoa số 8 sang lọ hoa số 2.
 - Chuyển bông hoa đang cắm ở lọ hoa số 2 sang lọ hoa số 1.
 - Chuyển bông hoa đang cắm ở lọ hoa số 1 sang lọ hoa số 5.
 - Chuyển bông hoa đang cắm ở lọ hoa số 5 sang lọ hoa số 6.
 - Do lọ hoa số 6 đang trống, luật này thành công và bông hoa số 9 được cắm vào lọ số 7.

Statement
is not
available
on
English
language

L. Xâu chữ số

0.5 seconds, 256 megabytes

Trong bài toán này, chúng ta chỉ quan tâm đến các xâu kí tự gồm toàn các chữ số từ 0 đến 9. Đồng thời, chúng ta chỉ xét một phép biến đổi sau: Đảo vị trí hai kí tự liên tiếp nhau trong xâu, nếu như đây là hai chữ số có chênh lệch lớn hơn 1.

Ví dụ, từ xâu 2217997 ta có thể biến đổi thành 2271997 (bằng việc đổi chỗ hai kí tự ở vị trí thứ ba và thứ tư), nhưng chúng ta không thể đổi chỗ hai kí tự ở vị trí thứ hai và thứ ba, vì hai chữ số 1 và 2 chỉ lệch nhau 1.

Ta nói hai xâu kí tự (chỉ gồm các chữ số) là *tương đương* khi và chỉ khi có thể biến đổi từ xâu này sang xâu kia bằng việc thực hiện hữu hạn lần phép biến đổi nói trên.

Bạn được cho một tập hợp các xâu chữ số S . Từ S , bạn xây dựng tập hợp các xâu chữ số T thỏa mãn: T chứa tất cả các hoán vị vòng quanh của các phần tử trong S . Ví dụ, nếu $S = \{227, 1997\}$ thì $T = \{227, 272, 722, 1997, 9971, 9719, 7199\}$. Hãy tìm cách chọn ra nhiều xâu chữ số nhất trong T sao cho các xâu chữ số này đôi một không tương đương.

Input

Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 50$) là số phần tử thuộc tập S .

Problems - Codeforces

Trong n dòng tiếp theo, mỗi dòng chứa một xâu chữ số thể hiện một phần tử của tập S . Dữ liệu vào đảm bảo các xâu kí tự chỉ gồm các chữ số từ 0 đến 9, và tổng độ dài các xâu kí tự trong S không quá 2500.

Output

In ra một số nguyên duy nhất là số xâu lớn nhất có thể chọn được từ T sao cho các xâu được chọn đôi một không tương đương.

input
1 017040
output
4

input
4 474 474 744 474
output
1

Trong ví dụ thứ nhất, ta có $T = \{017040, 170400, 704001, 040017, 400170, 001704\}$. Trong sáu xâu này, chỉ có hai cặp xâu chữ số tương đương là $(040017, 704001)$ và $(001704, 400170)$. Như vậy, ta có thể chọn được tối đa 4 xâu đôi một không tương đương.

Trong ví dụ thứ hai, ta có $T = \{447, 474, 744\}$ và các xâu này đôi một tương đương. Vì vậy ta chỉ chọn được tối đa 1 xâu.

Statement
is not
available
on
English
language

NAD1. Chiến dịch Hà Lan của Đức Quốc Xã

0.25 seconds, 32 megabytes

Tháng 5 năm 1940, sau khi kết thúc 8 tháng của cuộc *Chiến tranh buồn cười*, Đức Quốc Xã mở cuộc tấn công Hà Lan, mở đầu cho kế hoạch Vàng xâm chiếm Tây Âu trong mùa hè năm đó.

Cuộc tấn công đã thành công vang dội: quân đội Đức buộc Hà Lan phải đầu hàng chỉ sau 5 ngày tấn công với thương vong cực kỳ nhỏ. Điểm mấu chốt làm nên thắng lợi quan trọng này là chiến thuật mới mẻ của Đức: Lần đầu tiên trong lịch sử, quân nhảy dù được huy động để đánh chiếm các mục tiêu quan trọng tại Rotterdam, trước khi bộ binh Đức tiến vào đây.

Địa phận thành phố Rotterdam thuộc Hà Lan được mô phỏng bởi một bảng r hàng và c cột. Các hàng được đánh số từ 1 tới r , các cột được đánh số từ 1 tới c . Ô nằm trên hàng i , cột j được ký hiệu là (i, j) .

Trước cuộc chiến, các tình báo Đức thu thập thông tin về hệ thống phòng thủ của Hà Lan, và họ phát hiện ra n vị trí có lực lượng mỏng, đó là các ô $(x_1, y_1), (x_2, y_2), (x_n, y_n)$. Vì vậy, bộ chỉ huy không quân Đức đề ra kế hoạch tấn công như sau:

- Trong ngày đầu tiên, lính dù Đức sẽ đổ bộ vào n vị trí ít bố phòng như trên và chiếm các khu vực này.

- Trong các ngày tiếp theo, lính dù Đức sẽ tăng viện để tấn công và chiếm đóng các ô kề cạnh với những ô đã chiếm được ở những ngày trước đó.

Với kế hoạch như trên, giả sử rằng quân Đức đủ mạnh để chiếm đóng mọi vị trí muốn chiếm, hãy tính xem quân Đức mất bao nhiêu ngày để chiếm được toàn bộ thành phố Rotterdam.

Input

Dòng đầu tiên chứa ba số nguyên r, c và n ($1 \leq r, c \leq 10^2, 1 \leq n \leq 10^4$) — số hàng, số cột của lãnh thổ thành phố Rotterdam và số vị trí xung yếu.

n dòng tiếp theo, mỗi dòng chứa hai số nguyên x_i và y_i ($1 \leq x_i \leq r, 1 \leq y_i \leq c$) thể hiện một ô xung yếu.

Output

Một số nguyên duy nhất là số ngày cần thiết để Đức chiếm được Rotterdam theo kế hoạch dự định như trên.

input
3 4 2
2 2
3 4
output
3

Statement
is not
available
on
English
language

NAD2. ICPC

0.25 seconds, 16 megabytes

Kỳ thi Lập trình sinh viên quốc tế ICPC (*International Collegiate Programming Contest*) là kỳ thi lập trình lớn nhất và uy tín nhất dành cho khối sinh viên đại học trên toàn thế giới. Kể từ khi được thành lập vào năm 1970, kỳ thi ICPC đã thu hút hơn 50 ngàn sinh viên đến từ hơn 3 ngàn trường đại học trên thế giới tham gia tranh tài ở các vòng loại khu vực, để chọn ra 128 đội mạnh nhất thi đấu tại trận Chung kết thế giới (*ICPC World Finals*) diễn ra vào tháng 4 - 6 hàng năm. Tại Việt Nam, mùa thi ICPC được diễn ra từ tháng 10 tới tháng 12, gồm 5 vòng thi chính thức: 3 vòng loại 3 miền Bắc — Trung — Nam, 1 vòng loại quốc gia và 1 vòng loại khu vực châu Á. Phong trào ICPC tại Việt Nam ngày càng lớn mạnh và thu hút được hơn 700 sinh viên công nghệ thông tin và gần 100 học sinh chuyên tin trên cả nước trong năm 2019 vừa qua. Trường Đại Học Công Nghệ — Đại Học Quốc Gia Hà Nội là ngôi trường có thành tích xuất sắc áp đảo của cả nước, với 6 năm liên tiếp góp mặt tại vòng chung kết thế giới, và sở hữu ba đội tuyển Việt Nam có thứ hạng cao nhất trên thế giới từ trước tới nay: Đội *Unsigned* — xếp hạng 15 năm 2018, đội *Byte* — xếp hạng 28 năm 2016 và đội *Linux* — xếp hạng 34 năm 2017.

Problems - Codeforces

Theo thể lệ của kỳ thi ICPC, các thí sinh tranh tài theo hình thức đội 3 người, tham gia giải từ 10 đến 13 bài tập thuật toán trong vòng 300 phút (5 giờ). Khác với các kỳ thi OI ở cấp THPT, thí sinh chỉ được ghi nhận điểm khi giải trọn vẹn một bài (đúng hoàn toàn bộ test hay AC) chứ không được điểm nếu có một phần test giải đúng. Ngoài số bài giải được, thành tích của thí sinh còn được tính bởi chỉ số thời gian phạt (*penalty*) như sau: Mỗi bài giải được (AC) sẽ đóng góp lượng penalty bằng số phút kể từ đầu cuộc thi tới thời điểm AC đầu tiên của bài đó, cộng thêm 20 phút cho mỗi lần nộp sai trước đó. Tổng penalty của một đội bằng tổng lượng penalty do các bài mà đội đó giải được đóng góp. Các bài không giải được cũng như những lần nộp sai ở các bài này không làm tăng penalty.

Ví dụ, nếu một đội giải được 3 bài tại các thời điểm 22 phút, 7 phút và 97 phút; với số lần nộp sai lần lượt là 0, 1 và 2. Khi đó, tổng penalty của đội sẽ là $22 + 7 + 97 + 20 \cdot (0 + 1 + 2) = 186$.

Đội nào giải được nhiều bài hơn là đội có thành tích cao hơn. Trong các đội có cùng số bài giải được, đội nào có penalty thấp hơn là đội có thứ hạng cao hơn. Trong các đội có cùng số bài giải được và cùng penalty, đội nào có thời điểm giải xong bài cuối cùng nhỏ hơn là đội chiến thắng.

Bước vào kỳ thi ICPC, GS.PVH cùng đồng đội nhanh chóng đọc hết n bài trong đề. Ngay lập tức, GS nhận ra đội mình có thể làm hết cả n bài trong đề, nhưng để làm bài thứ i thì phải mất t_i phút và có thể không đủ thời gian làm hết các bài. Đồng thời, nhờ khả năng code cực chuẩn, GS tự tin rằng mình có thể AC ngay lần đầu tiên và không bao giờ chịu mất 20 phút penalty cho một lần nộp sai. Chỉ có một vấn đề duy nhất: Việc giải bất cứ bài nào phải cần cả ba thành viên cùng góp sức, do đó đội không thể làm việc trên hai bài trong cùng một thời điểm.

GS muốn tìm ra một tập hợp các bài để giải cùng thứ tự giải bài được chọn sao cho thành tích của đội là cao nhất: Số bài giải được là lớn nhất, penalty phải nhỏ nhất và thời điểm giải xong bài cuối cùng phải sớm nhất. Các bạn hãy giúp GS nhé.

Input

Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 26$) — số bài trong kỳ thi. Dòng thứ hai chứa n số nguyên t_1, t_2, \dots, t_n ($1 \leq t_i \leq 500$), trong đó t_i là thời gian cần thiết để đội của GS giải bài thứ i .

Output

Dòng đầu tiên ghi ra ba số nguyên $s p l$ — số bài đội của GS làm được, penalty của đội và thời điểm giải xong bài cuối cùng. Dòng thứ hai chứa một xâu ký tự gồm s ký tự in hoa phân biệt, thể hiện thứ tự các bài được giải từ đầu cuộc thi. Các bài được ký hiệu bằng các chữ cái in hoa phân biệt: Bài đầu tiên là bài 'A', bài thứ hai là bài 'B', bài thứ ba là bài 'C'... Nếu có nhiều phương án tối ưu, bạn cần in ra xâu ký tự có thứ tự từ điển nhỏ nhất.

input
5
45 60 75 90 105
output
4 600 270
ABCD

Statement
is not
available
on
English
language

NAD3. Bảng số

0.25 seconds, 9 megabytes

Chú ý giới hạn bộ nhớ đặc biệt của bài này.

Cho một bảng A gồm m hàng và n cột. Cần điền các số nguyên vào bảng A này sao cho:

- Các số được điền vào bảng có giá trị từ 1 tới $m \cdot n$.
- Mỗi giá trị nguyên dương từ 1 tới $m \cdot n$ xuất hiện đúng một lần trong bảng.
- Với hai ô (x_1, y_1) và (x_2, y_2) bất kỳ, nếu $x_1 + y_1 < x_2 + y_2$ thì $A_{x_1, y_1} < A_{x_2, y_2}$.

Cho hai số m và n , đếm số bảng A thỏa mãn điều kiện trên.

Input

Gồm một dòng chứa hai số nguyên dương duy nhất m và n .

Output

Gồm một số nguyên duy nhất là số cách điền bảng A thỏa mãn các điều kiện trên. Do số cách có thể rất bé, bạn cần in ra kết quả theo modulo $10^9 + 7$.

Scoring

- Subtask 1 (35 điểm): $m \leq 2, n \leq 10^6$
- Subtask 2 (40 điểm): $m \leq 10^6, n \leq 10^6$
- Subtask 3 (25 điểm): $m \leq 10^6, n \leq 10^9$

input
2 2
output
2

Hai cách điền thỏa mãn là

1 2

3 4

và

1 3

2 4