

Trại hè 2025 - TBTNPTLCQNHN - Lớp 11

A. Số còn thiếu

1 second, 512 megabytes

Cho bảng số hình chữ nhật kích thước $N \times M$ đã điền các số nhỏ hơn hoặc bằng $N \times M$.

Số còn thiếu của một hình chữ nhật là số nhỏ nhất không xuất hiện trong hình chữ nhật đó khi xét từ $1, 2, 3, \dots$

Cho hai số X, Y , hãy tính tích tất cả các số còn thiếu của mọi hình chữ nhật con kích thước $X \times Y$ của bảng số.

Input

Dòng đầu tiên gồm bốn số nguyên dương N, M, X, Y ($N \times M \leq 5 \times 10^5; X \leq N; Y \leq M$)

N dòng sau, mỗi dòng gồm M số mô tả các số trong bảng số hình chữ nhật.

Output

In ra một số nguyên duy nhất là số dư của tích tìm được chia cho $10^9 + 7$.

Scoring

- 1. 30%: $N, M \leq 50$
- 2. 50%: $N, M \leq 630$
- 3. 20%: không có ràng buộc gì thêm.

input
3 4 2 3 1 2 3 2 2 3 1 4 1 1 2 6
output
400

B. Trung vị

3 s, 1024 MB

Cho dãy hoán vị a có n phần tử, đếm số dãy con liên tiếp có phần tử đầu tiên là trung vị của dãy con đó. Trung vị của một dãy số m phần tử là: với m lẻ thì khi sắp xếp dãy này, phần tử ở giữa của dãy là trung vị, còn với m chẵn thì khi sắp xếp dãy, trung vị là trung bình cộng của hai phần tử ở giữa dãy.

Input

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 2 \cdot 10^5$).
- Dòng tiếp theo chứa n số nguyên a_i ($1 \leq a_i \leq n$), các số phân biệt.

Output

Ghi ra tổng số dãy con liên tiếp thỏa mãn.

input
5 1 2 3 4 5
output
5

input
6 3 2 1 6 4 5
output
8

input
7 2 1 5 3 6 7 4
output
11

C. Xếp lịch tư vấn ưu tiên

1 second, 256 megabytes

Bạn làm việc tại một trung tâm tư vấn khách hàng. Mỗi khi có một yêu cầu đặt lịch mới, hệ thống sẽ tự động thêm vào danh sách chờ theo thời gian thực.

Mỗi khách hàng có một chỉ số ưu tiên là số nguyên (có thể âm). Sau khi thêm mỗi yêu cầu mới, hệ thống sẽ thực hiện:

- Sắp xếp toàn bộ danh sách các khách theo mức độ ưu tiên tăng dần.
- Chọn ra các khách ở các vị trí $1, k + 1, 2k + 1, 3k + 1, \dots$
- Tính tổng các chỉ số ưu tiên của các khách này, gọi là *điểm tập trung phục vụ*.

Yêu cầu: Sau mỗi lần thêm lịch, hãy tính và in ra điểm tập trung phục vụ tương ứng.

Input

- Dòng đầu tiên chứa hai số nguyên n và k ($1 \leq n \leq 10^5, 1 \leq k \leq n$) — số lượt thêm yêu cầu và khoảng bước ưu tiên.
- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — chỉ số ưu tiên của từng yêu cầu theo thứ tự đến.

Output

In ra n dòng, dòng thứ i là tổng ưu tiên của các khách được chọn sau khi xử lý i yêu cầu đầu tiên.

Scoring

Subtask	Điểm	Ràng buộc
1	10	$n \leq 100$
2	10	$n \leq 10^5$ và $k = n$
3	30	$n \leq 10^5$ và $k = 2$
4	50	Không giới hạn thêm

input
6 3 5 1 6 1 4 7

output
5
1
1
7
6
6

D. Số lượng tổng

4 s, 256 megabytes

Cho dãy a_1, a_2, \dots, a_n . Có q truy vấn (l, r, x, y) , hãy đưa ra số lượng tổng khác nhau nằm trong khoảng $[x, y]$ mà các số a_l, a_{l+1}, \dots, a_r có thể sinh ra.

Input
Dòng đầu chứa số n, q

Dòng thứ hai chứa n số nguyên dương a_i

q dòng tiếp theo mỗi dòng chứa bốn số l, r, x, y mô tả các truy vấn.

Output
Mỗi truy vấn đưa kết quả trên một dòng.

Scoring
Trong tất cả các test:

- $1 \leq n \leq 500$
- $1 \leq q \leq 1\,500\,000$
- $a_i > 0$
- $S = \sum a_i \leq 250\,000$
- $1 \leq l \leq r \leq n$
- $1 \leq x \leq y \leq 1\,000\,000$

Subtasks:

- (5p): $q \leq 400$
- (10p): $n \leq 300; q, S \leq 90\,000$
- (10p): $q \leq 4\,000$
- (25p): $n \leq 400, q \leq 500\,000, S \leq 160\,000$
- (30p): $q \leq 500\,000$
- (20p): không có ràng buộc gì thêm.

input
5 2
12 3 4 6 8
2 4 6 15
1 5 1 1000
output
5
27

E. Cuộc đua xe

4 s, 1024 MB

Trên một đường đua là một đường thẳng, có k xe đua tự động xuất phát từ điểm 0 có vận tốc $\frac{1}{a_i}$, cần di chuyển qua m điểm đặc biệt, điểm thứ j có khoảng cách với điểm 0 là b_j . Cuộc đua diễn ra như sau:

- Mỗi xe đua có một điểm nhớ mà ban đầu chính là điểm 0
- Các xe cùng bắt đầu di chuyển từ điểm 0
- khí một hoặc nhiều xe đến một điểm đặc biệt thì xe có chỉ số nhỏ nhất trong số các xe đến được một điểm đặc biệt sẽ được thay đổi điểm

- nhớ thành điểm đặc biệt mà xe vừa đến, còn tất cả $k - 1$ xe còn lại sẽ bị quay trở lại điểm nhớ của xe đó
- Nếu một xe có điểm nhớ là điểm b_m thì được coi là hoàn thành và coi như không tham gia vào cuộc đua nữa
 - Cuộc đua kết thúc khi tất cả các xe đều hoàn thành.

Hãy cho biết tổng số lần bị quay lại điểm nhớ của tất cả các xe. Nhận thấy bài toán vẫn chưa đủ khó, bạn được cho n xe, và cần phải xét cuộc đua với k xe đầu tiên với k từ 1 đến n .

Input
Dòng đầu chứa số n, m ($2 \leq n \leq 150\,000; 1 \leq m \leq 150\,000$)

Dòng thứ hai chứa n số a_i ($1 \leq a_i \leq 10^9$)

Dòng thứ ba chứa m số b_j ($1 \leq b_1 < b_2 < \dots < b_m \leq 150\,000$)

Output
Hãy ra kết quả n số tương ứng với k từ 1 đến n .

- Scoring**
- (5p): $n = 2; m \leq 50$
 - (10p): $n, m \leq 50$
 - (10p): $n \leq 1000; m \leq 5$
 - (10p): $n, m \leq 10^5; b_j - b_{j-1} = b_1$
 - (5p): $n, m \leq 10^5; a_i$ bằng nhau
 - (10p): $n \leq 100; m \leq 10^5$
 - (5p): $n, m \leq 10^5; a_i \leq 2$
 - (5p): $n \leq 10^5; m = 2$
 - (10p): $n \leq 10^4; m \leq 10^5$
 - (10p): $n \leq 5 \cdot 10^4; m \leq 10^5$
 - (5p): $n, m \leq 10^5$
 - (5p): $n \leq 10^5$
 - (10p): không có ràng buộc gì thêm.

input
3 3
1 2 3
2 6 12
output
0
4
11

input
3 3
3 2 1
2 6 12
output
0
5
13

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

F. Đường đi trọng số liên tiếp

4 seconds, 512 megabytes

Cho đồ thị hai chiều n đỉnh m cạnh. Cạnh i nối hai đỉnh u_i, v_i có trọng số w_i . Hãy đếm số đường đi (có thể lặp lại đỉnh) thoả mãn trọng số của các cạnh đi qua tạo thành dãy số nguyên liên tiếp $x, x + 1, x + 2, \dots$

Input

Dòng đầu chứa số n, m ($2 \leq n \leq 5 \cdot 10^5; 1 \leq m \leq 10^6$)

m dòng tiếp theo, mỗi dòng chứa ba số u_i, v_i, w_i
($1 \leq u_i \neq v_i \leq n; 1 \leq w_i \leq 10^9$)

Output

Ghi ra số đường đi thoả mãn, ghi ra số dư cho $10^9 + 7$.

Scoring

- (20p): $n, m \leq 22$
- (30p): $n \leq 1000; m \leq 5000$
- (50p): không có ràng buộc gì thêm.

input
4 4
1 2 1
2 3 2
3 4 3
1 3 2
output
10

G. Đường đi trọng số tăng dần

2 seconds, 512 megabytes

Cho đồ thị đơn hai chiều n đỉnh đánh số từ 1 đến n và m cạnh đánh số từ 1 đến m . Cạnh thứ i nối hai đỉnh u_i với v_i có trọng số là w_i . Mỗi cặp đỉnh có nhiều nhất một cạnh nối, không có cạnh nào nối một đỉnh với chính nó. Hãy cho biết đường đi dài nhất từ đỉnh 1 đến đỉnh x ($1 \leq x \leq n$) thoả mãn trọng số của các cạnh đi qua là tăng dần.

Input

Dòng đầu chứa số n, m ($2 \leq n \leq 10^5; 1 \leq m \leq 10^6$)

m dòng tiếp theo, mỗi dòng chứa ba số u_i, v_i, w_i
($1 \leq u_i \neq v_i \leq n; 1 \leq w_i \leq 10^9$)

Output

Ghi ra một dòng gồm n số tương ứng là số cạnh nhiều nhất đi qua được để đi từ 1 đến x . Nếu không thể đi đến được x thì ghi ra 0.

input
8 12
1 5 1
5 6 2
1 6 4
7 8 5
6 2 6
2 8 7
5 8 8
1 4 9
3 5 10
3 7 12
3 4 11
5 7 14
output
3 3 6 7 8 2 7 4

H. Đồ thị

1 second, 512 megabytes

Cho đồ thị G có hướng không chu trình, có n đỉnh đánh số từ 1 đến n . Đếm số đường đi từ 1 đến n sao cho đi qua tất cả p đỉnh theo đúng thứ tự đưa ra.

Input

- Dòng đầu tiên chứa n, m, p ($1 \leq n, m, p \leq 10^6$) là số đỉnh, số cạnh và số đỉnh phải đi qua.
- Dòng thứ hai chứa p số là các đỉnh phải đi qua.
- m dòng tiếp theo mỗi dòng chứa u, v là cạnh từ u đến v .

Output

Đưa ra kết quả lấy số dư cho $10^9 + 7$.

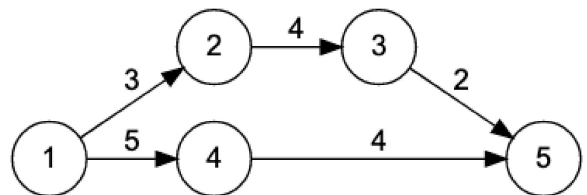
input
6 9 3
3 5 6
1 3
1 3
1 2
2 3
2 4
3 4
3 5
4 5
5 6
output
6

I. Đoàn tụ

2 seconds, 512 megabytes

Sau một thời gian dài xa cách, Alice và Bob đã có thể đoàn tụ. Họ sống trong một quốc gia có n thành phố, được đánh số từ thành phố 1 đến thành phố n . Bob lái xe từ nhà của anh ấy ở thành phố 1 đến nhà của Alice ở thành phố n .

Khi Alice hỏi Bob anh đã đi qua những thành phố nào, anh sững sờ khi mình không nhớ gì cả. Bob tối ưu và lái xe không dừng lại, nên ta biết rằng không có con đường nào nhanh hơn con đường anh ấy đã đi. Anh ấy cũng có một chương trình lưu lại các thông tin di chuyển. Mỗi khi Bob lái xe qua một thành phố, chương trình này sẽ ghi lại thời gian từ khi anh rời thành phố 1 đến khi anh đến thành phố hiện tại.



Trong ví dụ trên, có hai con đường nhanh nhất có thể mà Bob có thể đi từ thành phố 1 đến thành phố n : $1 - 2 - 3 - 5$ hoặc $1 - 4 - 5$. Cả hai con đường mất tổng cộng 9 đơn vị thời gian. Đường dẫn đầu tiên sẽ có thông tin lưu lại là 0, 3, 7, 9 và đường thứ hai sẽ là 0, 5, 9.

Không may, bộ nhớ trong chương trình lưu trữ bị hỏng. Bob nghĩ rằng một số thông tin đã biến mất, và những thông tin còn lại bị xáo trộn một cách tùy ý. Với những gì còn lại của chương trình, bạn có thể tạo lại đường dẫn của Bob không?

Input

Dòng đầu chứa số n, m, d ($2 \leq n \leq 2 \times 10^5; 1 \leq m \leq 3 \times 10^5; 1 \leq d \leq n$) là số thành phố, số đoạn đường một chiều kết nối hai thành phố, và số thông tin còn lại trong chương trình.

m dòng tiếp theo mỗi dòng chứa hai số u, v, w ($1 \leq w \leq 10^6$) mô tả đoạn đường đi từ u đến v có thời gian di chuyển là w . Tồn tại ít nhất một đường đi từ 1 tới n và có thể có nhiều đoạn đường kết nối hai thành phố.

d dòng tiếp theo mỗi dòng chứa một số nguyên t ($0 \leq t \leq 10^{18}$ là thời điểm di chuyển từ thành phố 1 tới một thành phố trên đường đi từ 1 tới n . Dữ liệu đảm bảo các số phân biệt.

Output

Ghi ra 0 nếu không tồn tại cách di chuyển như mô tả.

Ghi ra 1 nếu có rất nhiều các đi.

Ngược lại, dòng đầu tiên in ra số lượng thành phố đã đi qua, tiếp theo liệt kê các thành phố này trên các dòng.

input
5 5 2 1 2 3 2 3 4 3 5 2 1 4 5 4 5 4 5 9
output
3 1 4 5

input
6 8 2 1 2 1 2 3 2 3 6 8 1 4 3 4 5 4 5 6 4 5 2 7 1 6 13 0 3
output
1

input
2 1 1 1 2 10 5
output
0

J. Chu trình nhỏ nhất

1 second, 512 megabytes

Cho đồ thị vô hướng n đỉnh m cạnh, đếm số chu trình có kích thước nhỏ nhất (kích thước của chu trình là số đỉnh thuộc chu trình đó).

Input

Dòng đầu chứa số n, m ($3 \leq n \leq 3000, 3 \leq m \leq 6000$)

Problems - Codeforces

m dòng tiếp theo mô tả các cạnh của đồ thị, dữ liệu đảm bảo không có hai đỉnh nào có nhiều hơn một cạnh nối, không có cạnh nối một đỉnh với chính nó. Dữ liệu đảm bảo luôn tồn tại ít nhất một chu trình.

Output

Ghi ra số lượng chu trình có kích thước nhỏ nhất.

input
4 4 1 2 2 3 3 4 4 1
output
1

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

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

K. Đường đi ngắn nhất

2 seconds, 512 megabytes

Cảnh sát đang cần sự giúp đỡ của bạn trong việc tìm kiếm nơi trú ở của một tên tội phạm, kẻ đang ẩn náu ở đâu đó trong thành phố gồm n địa điểm khác nhau, được đánh số từ 1 đến n , và có m đường hai chiều kết nối hai địa điểm khác nhau.

Thông tin từ các người dân trong thành phố, cảnh sát biết được tên tội phạm đã di chuyển từ một địa điểm x nào đó để đến nơi trú ẩn y nào đó. Và các nhân chứng có trông thấy hắn xuất hiện ở k địa điểm u_1, u_2, \dots, u_k trên đường đi từ x đến y theo thứ tự nào đó mà cảnh sát chưa xác định. Lưu ý rằng tên tội phạm có thể di chuyển từ x đến y thông qua các địa điểm khác không có trong k địa điểm mà các nhân chứng trông thấy. Tuy nhiên, cảnh sát đã phân tích đặc điểm của tên tội phạm này và biết rằng hắn sẽ luôn di chuyển theo đường đi ngắn nhất giữa hai địa điểm x và y .

Nhiệm vụ của bạn là tìm các địa điểm có thể là y để giúp cảnh sát có thể nhanh chóng tìm được hắn. Tất nhiên, có thể lời khai của các nhân chứng không đồng nhất dẫn đến không tìm thấy một địa điểm y nào thỏa mãn.

Input

Dòng đầu chứa số nguyên n, m, k (

$$1 \leq k \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 2 \cdot 10^5)$$

Dòng thứ hai chứa k số nguyên u_i mô tả những địa điểm tên tội phạm đi qua ($1 \leq u_i \leq n$)

m dòng tiếp theo, mỗi dòng chứa ba số nguyên a, b, c mô tả có đường nối trực tiếp giữa hai địa điểm a và b có độ dài là c (

$$1 \leq a \neq b \leq n, 1 \leq c \leq 10^9)$$

Output

Dòng đầu ghi ra số p là số địa điểm có thể là y

Dòng thứ hai ghi ra p số nguyên là những địa điểm có thể là y theo thứ tự tăng dần.

Scoring

- 15% số điểm có $m = n - 1$, và mỗi địa điểm được kết nối trực tiếp với tối đa 2 địa điểm khác.
- 15% số điểm có $m = n - 1$
- 15% số điểm có $n, m \leq 100$
- 15% số điểm có $n, m \leq 1000$
- 20% số điểm có $k \leq 5$
- 20% số điểm còn lại không có ràng buộc gì thêm

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

[Codeforces](#) (c) Copyright 2010-2025 Mike Mirzayanov
The only programming contests Web 2.0 platform