

Problem A. Color query

Time Limit 2250 ms

Mem Limit 524288 kB

Given a sequence of n elements a_1, a_2, \dots, a_n and a number of queries d . A query d is a pair (i, j) ($1 \leq i \leq j \leq n$). For each query $d(i, j)$, you need to return the number of distinct elements in the subarray a_i, a_{i+1}, \dots, a_j .

Input

- Line 1: n ($1 \leq n \leq 30000$).
- Line 2: n numbers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$).
- Line 3: q ($1 \leq q \leq 200000$), the number of queries d .
- In the next q lines, each line contains 2 numbers i, j representing a query d ($1 \leq i \leq j \leq n$).

Output

- For each query $d(i, j)$, print the number of distinct elements in the subarray a_i, a_{i+1}, \dots, a_j on a new line.

Input	Output
5 1 1 2 1 3 3 1 5 2 4 3 5	3 2 3

Problem B. Chia đề trị cơ bản

Time Limit 1000 ms

Mem Limit 524288 kB

Cho một dãy n phần tử a_1, a_2, \dots, a_n và một số các truy vấn-k. Một truy vấn-k là một bộ ba (i, j, k) ($1 \leq i \leq j \leq n$). Với mỗi truy vấn-k (i, j, k) , bạn phải trả về số phần tử lớn hơn k nằm trong dãy con a_i, a_{i+1}, \dots, a_j .

Input

- Dòng 1: n ($1 \leq n \leq 30000$).
- Dòng 2: n số a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).
- Dòng 3: q ($1 \leq q \leq 200000$), số truy vấn-k.
- Trong q dòng tiếp theo, mỗi dòng chứa 3 số i, j, k thể hiện một truy vấn-k ($1 \leq i \leq j \leq n, 1 \leq k \leq 10^9$).

Output

- Với mỗi truy vấn-k (i, j, k) , in ra số phần tử lớn hơn k trong dãy con a_i, a_{i+1}, \dots, a_j trên một dòng.

Input	Output
5 5 1 2 3 4 3 2 4 1 4 4 4 1 53 1000	2 0 3

Problem C. K-query II

Time Limit 1000 ms

Mem Limit 262144 kB

Sắp xếp là một trong những thuật toán phổ biến nhất trong khoa học máy tính. An vừa được giới thiệu các thuật toán sắp xếp và biết rằng thuật toán sắp xếp nhanh (Quick Sort) có độ phức tạp trung bình là $O(n \log n)$. Tuy nhiên An vẫn muốn sáng tạo một thuật toán sắp xếp mới và xem thử liệu nó có tốt hơn Quick Sort. Xét một mảng n số nguyên, mỗi phần tử có giá trị từ 1 đến n và mỗi giá trị chỉ xuất hiện một lần trong mảng. Thuật toán này sẽ sắp xếp mảng trên trong n bước như sau:

- Bước 1: phần tử có giá trị 1 sẽ được chuyển đến vị trí 1 bằng cách liên tiếp hoán đổi vị trí với phần tử liền kề với nó.
- Bước 2: phần tử có giá trị n sẽ được chuyển đến vị trí n bằng cách như bước 1.
- Bước 3: phần tử có giá trị 2 sẽ được chuyển đến vị trí 2 bằng cách tương tự.
- Bước 4: phần tử có giá trị $n - 1$ sẽ được chuyển đến vị trí $n - 1$ bằng cách tương tự.
- Và cứ thế tiếp tục đến bước n .

Tổng quát hơn, ở bước lẻ, An sẽ chọn phần tử có giá trị nhỏ nhất chưa được sắp xếp và di chuyển phần tử này đến vị trí đúng của nó. Ở bước chẵn An sẽ chọn phần tử có giá trị lớn nhất chưa được sắp xếp và di chuyển phần tử này đến vị trí đúng của nó.

Yêu cầu: Viết chương trình cho biết số lần hoán đổi vị trí các phần tử tại mỗi bước của thuật toán trên.

Input

- Dòng đầu tiên chứa một số nguyên n cho biết số lượng phần tử của mảng.
- Trong n dòng tiếp theo, dòng thứ i chứa một số nguyên a_i ($1 \leq a_i \leq n$) cho biết giá trị của phần tử thứ i trong mảng. Các giá trị a_i không lặp lại.

Output

- Gồm n số nguyên, số thứ i cho biết số lần hoán đổi vị trí các phần tử ở bước thứ i của thuật toán. Mỗi số ghi trên một dòng.

Scoring

Subtask	Điểm	Giới hạn
1	50%	$n \leq 1000$
2	20%	$n \leq 10000$
3	30%	$n \leq 100000$

Input		Output	
7		5	
5		2	
2		1	
6		2	
7		1	
3		1	
1		0	
4			

Problem D. Chú voi con ở Bản Đôn

Time Limit 4000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Con voi nhỏ thích chơi với mảng. Anh ấy có mảng a , bao gồm n số nguyên dương, được đánh chỉ mục từ 1 đến n . Hãy ký hiệu số có chỉ mục i là a_i .

Ngoài ra, con voi nhỏ có m truy vấn đến mảng, mỗi truy vấn được đặc trưng bởi một cặp số nguyên l_j và r_j ($1 \leq l_j \leq r_j \leq n$). Đối với mỗi truy vấn l_j, r_j , con voi nhỏ phải đếm xem có bao nhiêu số x tồn tại, sao cho số x xuất hiện chính xác x lần trong các số $a_{l_j}, a_{l_j+1}, \dots, a_{r_j}$.

Giúp con voi nhỏ đếm số câu trả lời cho tất cả các truy vấn.

Nhập

Dòng đầu tiên chứa hai số nguyên cách biệt bởi dấu cách n và m ($1 \leq n, m \leq 10^5$) — kích thước của mảng a và số lượng truy vấn đến nó. Dòng tiếp theo chứa n số nguyên dương cách biệt bởi dấu cách a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$). Tiếp theo là m dòng chứa mô tả của các truy vấn, mỗi dòng một truy vấn. Dòng thứ j chứa mô tả của truy vấn thứ j dưới dạng hai số nguyên cách biệt bởi dấu cách l_j và r_j ($1 \leq l_j \leq r_j \leq n$).

Đầu ra

Trong m dòng in m số nguyên — câu trả lời cho các truy vấn. Dòng thứ j nên chứa câu trả lời cho truy vấn thứ j .

Ví dụ 1

Input	Output
7 2 3 1 2 2 3 3 7 1 7 3 4	3 1

Problem E. Bò sữa Vinamilk

Time Limit 2000 ms

Mem Limit 262144 kB

Liên đoàn Bò sữa của bác John (UCFG) đang chuẩn bị phái một đoàn đại biểu tới International bOvine olympIad* (IOI).

Có N chú bò sẽ tham gia cuộc tuyển chọn cho đoàn đại biểu này ($1 \leq N \leq 2 \cdot 10^5$). Chúng đang xếp thành một hàng, với chú bò i có giống là b_i .

Một đoàn đại biểu sẽ bao gồm một dãy bò liên tiếp có ít nhất hai con – hay nói cách khác, là những chú bò có số hiệu trong khoảng $[l, r]$ với hai số nguyên l và r thỏa mãn $1 \leq l < r \leq N$. Hai chú bò ở hai đầu của dãy đã được chọn sẽ làm "đội trưởng". Để tránh phối giống cận huyết, mỗi đội trưởng phải có giống khác với những chú bò còn lại (đội trưởng hay không là đội trưởng).

Hãy giúp UCFG đếm (cho mục đích đóng thuế) số lượng cách mà họ có thể chọn một đoàn đại biểu để phái tới IOI.

*International bOvine olympIad: Olympiad Bò Quốc tế.

Input

Dòng đầu tiên chứa số N .

Dòng thứ hai chứa N số nguyên b_1, b_2, \dots, b_N , trong khoảng $[1, N]$.

Output

Ghi ra số lượng đoàn đại biểu có thể phái đi, trên một dòng.

Input	Output
7 1 2 3 4 3 2 5	13

Giải thích

Các cặp đội trưởng của các đoàn đại biểu thỏa mãn là:

$(1, 2), (1, 3), (1, 4), (1, 7), (2, 3), (2, 4), (3, 4), (4, 5), (4, 6), (4, 7), (5, 6), (5, 7), (6, 7)$.

Ràng buộc

- Các test 1-3 thỏa mãn $N \leq 100$.
 - Các test 4-8 thỏa mãn $N \leq 5000$.
 - Các test 9-20 thỏa mãn $N \leq 200000$.
-

Problem F. Chimera Ant

Time Limit 1000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Mole đói rồi. Nó tìm thấy một tổ kiến, gồm n con kiến, được sắp xếp thành một hàng. Mỗi con kiến i ($1 \leq i \leq n$) có sức mạnh là s_i .

Để bữa tối của mình thêm thú vị, Mole tổ chức một phiên bản của trò chơi "Hunger Games" cho các con kiến. Nó chọn hai số l và r ($1 \leq l \leq r \leq n$) và mỗi cặp con kiến với chỉ số từ l đến r (bao gồm cả hai đầu) sẽ chiến đấu. Khi hai con kiến i và j chiến đấu, con kiến i chỉ nhận được một điểm chiến thắng nếu s_i là ước của s_j (tương tự, con kiến j chỉ nhận được một điểm chiến thắng nếu s_j là ước của s_i).

Sau khi tất cả các trận đấu đã kết thúc, Mole xếp hạng. Một con kiến i , với v_i điểm chiến thắng, sẽ được thả ra chỉ nếu $v_i = r - l$, hoặc nói cách khác chỉ nếu nó đã giành được một điểm trong mỗi trận đấu mà nó tham gia. Sau đó, Mole ăn phần còn lại của các con kiến. Lưu ý rằng có thể có nhiều con kiến được thả ra hoặc thậm chí không có con kiến nào được thả ra.

Để chọn ra chuỗi tốt nhất, Mole cho bạn t đoạn $[l_i, r_i]$ và yêu cầu cho mỗi đoạn đó có bao nhiêu con kiến sẽ bị ăn nếu những con kiến đó chiến đấu.

Nhập

Dòng đầu tiên chứa một số nguyên n ($1 \leq n \leq 10^5$), kích thước của tổ kiến.

Dòng thứ hai chứa n số nguyên s_1, s_2, \dots, s_n ($1 \leq s_i \leq 10^9$), là sức mạnh của các con kiến.

Dòng thứ ba chứa một số nguyên t ($1 \leq t \leq 10^5$), số lượng trường hợp kiểm tra.

Mỗi trong t dòng tiếp theo chứa hai số nguyên l_i và r_i ($1 \leq l_i \leq r_i \leq n$), mô tả một truy vấn.

Đầu ra

In ra đầu ra chuẩn t dòng. Dòng thứ i chứa số lượng con kiến mà Mole ăn từ đoạn $[l_i, r_i]$.

Ví dụ

Input	Output
5	4
1 3 2 4 2	4
4	1
1 5	1
2 5	
3 5	
4 5	

Ghi chú

Trong trận đấu thử đầu tiên, điểm chiến thắng cho mỗi con kiến là $v = [4, 0, 2, 0, 2]$, vì vậy con kiến số 1 được thả ra. Mole ăn các con kiến 2, 3, 4, 5.

Trong trường hợp thử thứ hai, điểm chiến thắng là $v = [0, 2, 0, 2]$ cho mỗi con kiến, vì vậy không có con kiến nào được thả ra và tất cả đều bị Mole ăn.

Trong trường hợp thử thứ ba, điểm chiến thắng là $v = [2, 0, 2]$ cho mỗi con kiến, vì vậy con kiến số 3 và 5 được thả ra. Mole chỉ ăn con kiến 4.

Trong trường hợp thử thứ tư, điểm chiến thắng là $v = [0, 1]$ cho mỗi con kiến, vì vậy con kiến số 5 được thả ra. Mole ăn con kiến 4.

Problem G. Con ngựa thành Troy

Time Limit 5000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Người La Mã đã tấn công lại. Lần này họ nhiều hơn người Ba Tư nhưng Shapur sẵn sàng để đánh bại họ. Anh ấy nói: "Một con sư tử không bao giờ sợ một trăm con cừu".

Tuy nhiên Shapur phải tìm ra điểm yếu trong quân đội La Mã để đánh bại họ. Vì vậy anh ấy đưa ra một số điểm yếu cho quân đội.

Theo ý kiến của Shapur, điểm yếu của một quân đội bằng số lượng bộ ba i, j, k sao cho $i < j < k$ và $a_i > a_j > a_k$ trong đó a_x là sức mạnh của người đứng ở vị trí x . Quân đội La Mã có một đặc điểm đặc biệt - sức mạnh của tất cả mọi người trong đó đều khác nhau.

Giúp Shapur tìm ra mức độ yếu của người La Mã.

Nhập

Dòng đầu tiên của đầu vào chứa một số nguyên n ($3 \leq n \leq 10^6$) - số người trong quân đội La Mã. Dòng tiếp theo chứa n số nguyên dương khác nhau a_i ($1 \leq i \leq n$, $1 \leq a_i \leq 10^9$) - sức mạnh của những người trong quân đội La Mã.

Đầu ra

Một số nguyên duy nhất, mức độ yếu của quân đội La Mã.

Vui lòng không sử dụng chỉ số `%lld` để đọc hoặc viết số nguyên 64-bit trong C++. Nên sử dụng `cout` (bạn cũng có thể sử dụng `%I64d`).

Ví dụ

Input	Output
3 3 2 1	1

Input	Output
3 2 3 1	0

Input	Output
4 10 8 3 1	4

Input	Output
4 1 5 4 3	1

Problem H. J97

Time Limit 2000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Cho hai dãy $a_1, a_2, a_3, \dots, a_n$ và $b_1, b_2, b_3, \dots, b_n$ có cùng độ dài n . Bạn hãy thực hiện m truy vấn thuộc một trong hai loại sau:

- $1 \ x \ y \ k$: Gán $b_{y+q} := a_{x+q}$ với mọi q ($0 \leq q < k$). Dữ liệu đảm bảo tất cả các phần tử cần sử dụng trong thao tác này đều có mặt ở hai dãy a và b .
- $2 \ x$: Tính giá trị của b_x .

Input

Dòng thứ nhất chứa hai số nguyên dương n, m ($n, m \leq 10^5$).

Dòng thứ hai chứa n số nguyên $a_1, a_2, a_3, \dots, a_n$ ($|a_i| \leq 10^9$).

Dòng thứ ba chứa n số nguyên $b_1, b_2, b_3, \dots, b_n$ ($|b_i| \leq 10^9$).

m dòng tiếp theo, mỗi dòng có dạng " $1 \ x \ y \ k$ " ($1 \leq x, y, k \leq n$) hoặc " $2 \ x$ " ($1 \leq x \leq n$) miêu tả một truy vấn thuộc một trong hai loại đã nêu ở đề bài.

Output

Với mỗi truy vấn loại 2, in ra kết quả trên một dòng.

Examples

Input	Output
5 10 1 2 0 -1 3 3 1 5 -2 0 2 5 1 3 3 3 2 5 2 4 2 1 1 2 1 4 2 1 2 4 1 4 2 1 2 2	0 3 -1 3 2 3 -1

Problem I. Xô số

Time Limit 2500 ms

Mem Limit 524288 kB

Một lần, Leha tìm thấy trong túi bên trái một mảng gồm n số nguyên, và trong túi bên phải q truy vấn có dạng $l\ r\ k$. Nếu có truy vấn, thì chúng phải được trả lời. Câu trả lời cho truy vấn là số nhỏ nhất x sao cho x xuất hiện trong khoảng $l\ r$ nhiều hơn $\frac{r-l+1}{k}$ lần hoặc -1 nếu không có số nào như vậy. Giúp Leha với nhiệm vụ khó khăn này.

Nhập vào

Dòng đầu tiên của dữ liệu đầu vào chứa hai số nguyên n và q ($1 \leq n, q \leq 3 \cdot 10^5$) — số lượng phần tử trong mảng và số lượng truy vấn tương ứng.

Dòng tiếp theo chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — mảng của Leha.

Mỗi dòng tiếp theo q chứa ba số nguyên l, r và k ($1 \leq l \leq r \leq n, 2 \leq k \leq 5$) — mô tả các truy vấn.

Đầu ra

Xuất câu trả lời cho mỗi truy vấn trên một dòng mới.

Ví dụ

Input	Output
4 2 1 1 2 2 1 3 2 1 4 2	1 -1

Input	Output
5 3 1 2 1 3 2 2 5 3 1 2 3 5 5 2	2 1 2

Problem H

Hard Queries

Lokk loves data structure, and his favorite leisure activity is inventing new and exotic data structure problems, and challenge his friends to solve it. Today, Lokk came up with the problem below and sent it to Quang.

You are given an array a containing n elements a_1, a_2, \dots, a_n . Your task is to process q queries of the following types:

- 1 p : Swap two elements a_p and a_{p+1} .
- 2 $l\ r\ x$: Calculate the sum of square of all positions from l to r whose element is equal to x . Formally, calculate $\sum_{i=l}^r i^2 \times [a_i = x]$, where $[P]$ denotes an expression that evaluates to 1 if P is true, and 0 otherwise.
- 3 $l\ r\ x$: Define b as the array of all positions in array a whose element is equal to x , listed from 1 to n . Calculate $\sum_{i=l}^r b_i^2$.

Being a brilliant problem solver, Quang tackled the problem quickly. Therefore, Lokk put an additional challenge: Quang has to answer the queries **online**, meaning that the queries are encoded in a way that the correct answer to the current query is required to decode the next one.

This is too much for Quang to handle, so he gave up on the problem and ask the contestants to help him. Can you solve it?

Input

The first line contains two integers n and q ($2 \leq n \leq 2 \times 10^5$, $1 \leq q \leq 2 \times 10^5$) — the length of a , and the number of queries to process.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — the elements of array a .

Each of the next q lines describe a query to process. The first number is t ($1 \leq t \leq 3$) — the type of the query. The given queries will be encoded in the following way: let $last$ be the answer to the last query of the type 2 or 3 that you have answered (initially, $last = 0$).

If $t = 1$, an integer p' follows ($1 \leq p' < n$). Perform the operation $p = (p' + last - 1) \bmod (n - 1) + 1$ to decode the query.

If $t = 2$ or $t = 3$, three integers l' , r' , and x' follow ($1 \leq l', r', x' \leq n$). Perform the following operations to decode the query:

- $l = (l' + last - 1) \bmod n + 1$
- $r = (r' + last - 1) \bmod n + 1$
- $x = (x' + last - 1) \bmod n + 1$
- If $l > r$, swap l and r .

For $t = 3$, it is guaranteed that r will not exceed the length of array b (defined in the query description).

Output

For each query of type 2 or 3, print a single integer — the answer to the query.

Sample explanation

Query 1 after decoding is 2 1 6 1. The positions from $l = 1$ to $r = 6$ whose element is equal to $x = 1$ are 1, 3 and 6. Therefore, the answer to this query is $1^2 + 3^2 + 6^2 = 46$.

Query 2 after decoding is 3 2 3 1. In this case, array b is $[1, 3, 6]$, so the answer to this query is $3^2 + 6^2 = 45$.

Query 3 after decoding is 2 6 6 2. Since $a_6 = 1$ which is different from $x = 2$, the answer to this query is 0.

Query 4 after decoding is 2 1 7 5. Since none of the elements of a equal $x = 5$, the answer to this query is 0.

Query 5 after decoding is 1 6. After swapping a_p and a_{p+1} (which is a_6 and a_7), array a become $[1, 2, 1, 3, 2, 2, 1]$.

Query 6 after decoding is 2 1 6 1. The answer to this query is $1^2 + 3^2 = 10$.

Query 7 after decoding is 3 2 3 1. The answer to this query is $3^2 + 7^2 = 58$.

Query 8 after decoding is 2 6 6 2. The answer to this query is $6^2 = 36$.

Sample Input 1

7 8	46
1 2 1 3 2 1 2	45
2 1 6 1	0
3 6 5 4	0
2 3 3 6	10
2 1 7 5	58
1 6	36
2 6 1 1	
3 7 6 5	
2 4 4 7	

Sample Output 1

Problem K. Cây tiền tố

Time Limit 1000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Chúng ta sẽ gọi một mảng gồm n số nguyên không âm $a[1], a[2], \dots, a[n]$ *thứ vị*, nếu nó thỏa mãn m các ràng buộc. Ràng buộc thứ i trong số m ràng buộc bao gồm ba số nguyên l_i, r_i, q_i ($1 \leq l_i \leq r_i \leq n$) có nghĩa là giá trị $a[l_i] \& a[l_i + 1] \& \dots \& a[r_i]$ phải bằng q_i .

Nhiệm vụ của bạn là tìm bất kỳ mảng *thứ vị* nào có n phần tử hoặc tuyên bố rằng mảng như vậy không tồn tại.

Biểu thức $x \& y$ có nghĩa là phép AND theo bit của các số x và y . Trong các ngôn ngữ lập trình C++, Java và Python, phép toán này được biểu diễn là "&", trong Pascal — là "and".

Nhập vào

Dòng đầu tiên chứa hai số nguyên n, m ($1 \leq n \leq 10^5, 1 \leq m \leq 10^5$) — số lượng phần tử trong mảng và số lượng giới hạn.

Mỗi dòng tiếp theo m chứa ba số nguyên l_i, r_i, q_i ($1 \leq l_i \leq r_i \leq n, 0 \leq q_i < 2^{30}$) mô tả giới hạn thứ i .

Xuất ra

Nếu mảng *thứ vị* tồn tại, trong dòng đầu tiên in "CÓ" (không có dấu ngoặc kép) và trong dòng thứ hai in n số nguyên $a[1], a[2], \dots, a[n]$ ($0 \leq a[i] < 2^{30}$) mô tả mảng *thứ vị*. Nếu có nhiều câu trả lời, in bất kỳ câu nào trong số đó.

Nếu mảng *thứ vị* không tồn tại, in "KHÔNG" (không có dấu ngoặc kép) trong một dòng duy nhất.

Ví dụ

Input	Output
3 1 1 3 3	YES 3 3 3

Input	Output
3 2 1 3 3 1 3 2	NO

Problem L. Convolutional Neural Network

Time Limit 1000 ms

Mem Limit 262144 kB

Mô tả đề bài

Ngân là một thành viên trong team tổ chức sự kiện tại VNG, sắp tới Ngân sẽ có rất nhiều sự kiện cần tham gia. Là một cô gái điệu đà, Ngân luôn thích trang điểm thật xinh khi tham gia các sự kiện này. Có n sự kiện và m cách lựa chọn thời gian trang điểm. Sự kiện thứ i sẽ có thời điểm bắt đầu là a_i và thời điểm kết thúc là b_i ($1 \leq i \leq n$). Cách trang điểm j sẽ có hiệu quả trong khoảng thời gian từ thời điểm c_j đến thời điểm d_j ($1 \leq j \leq m$). Nhiệm vụ của bạn là giúp Ngân tìm xem cách trang điểm nào giúp Ngân tham gia được nhiều sự kiện nhất trong m cách trên.

Lưu ý: Ngân chỉ có thể tham gia những sự kiện mà thời gian bắt đầu lẫn kết thúc đều nằm trong khoảng thời gian mà cách trang điểm Ngân chọn có thể duy trì hiệu quả. Ví dụ Ngân chọn cách trang điểm có hiệu quả từ thời điểm c_j đến d_j thì Ngân chỉ có thể tham gia những sự kiện có thời điểm tham gia và kết thúc là a_i và b_i sao cho $c_j \leq a_i \leq b_i \leq d_j$

Giới hạn: $1 \leq n, m \leq 10^5$, $1 \leq a_i, b_i \leq 10^9$, $1 \leq c_j, d_j \leq 10^9$.

Input

Dòng đầu tiên chứa 2 số nguyên n và m .

n dòng tiếp theo mỗi dòng chứa 2 số nguyên a_i và b_i .

m dòng tiếp theo mỗi dòng chứa 2 số nguyên c_j và d_j .

Output

In ra số sự kiện nhiều nhất mà Ngân có thể tham gia bằng cách chọn 1 trong những cách trang điểm được cho.

Input	Output
4 1 1 3 1 2 3 3 4 5 1 3	3

Giải thích

Ở đây Ngân chỉ có một cách trang điểm duy nhất và cách trang điểm này có hiệu quả từ thời điểm 1 đến thời điểm 3 nên Ngân có thể dùng nó để tham gia sự kiện 1, 2 và 3.

Problem M. Cây tiền tố 2

Time Limit 4000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Bạn có một mảng a , bao gồm n số nguyên a_1, a_2, \dots, a_n . Bạn được phép thực hiện hai phép toán trên mảng này:

1. Tính tổng các phần tử hiện tại trên đoạn $[l, r]$, nghĩa là, đếm giá trị

$$a_l + a_{l+1} + \dots + a_r.$$

2. Áp dụng phép xor với một số cho trước x cho mỗi phần tử trong mảng trên đoạn

$[l, r]$, nghĩa là, thực hiện $a_l = a_l \oplus x, a_{l+1} = a_{l+1} \oplus x, \dots, a_r = a_r \oplus x$. Phép toán này thay đổi chính xác $r - l + 1$ phần tử trong mảng.

Biểu thức $x \oplus y$ có nghĩa là áp dụng phép toán xor bit cho các số x và y . Phép toán cho trước tồn tại trong tất cả các ngôn ngữ lập trình hiện đại, ví dụ trong ngôn ngữ C++ và Java nó được đánh dấu là " \wedge ", trong Pascal — là " xor ".

Bạn có một danh sách m phép toán của loại đã chỉ định. Nhiệm vụ của bạn là thực hiện tất cả các phép toán cho trước, đối với mỗi truy vấn tổng bạn nên in kết quả bạn nhận được.

Nhập

Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) — kích thước của mảng. Dòng thứ hai chứa các số nguyên cách nhau bởi dấu cách a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^6$) — mảng gốc.

Dòng thứ ba chứa số nguyên m ($1 \leq m \leq 5 \cdot 10^4$) — số lượng phép toán với mảng. Dòng thứ i của m dòng tiếp theo đầu tiên chứa một số nguyên t_i ($1 \leq t_i \leq 2$) — loại của truy vấn thứ i .

Nếu $t_i = 1$, thì đây là truy vấn của tổng, nếu $t_i = 2$, thì đây là truy vấn để thay đổi các phần tử của mảng. Nếu phép toán thứ i là loại 1, thì tiếp theo là hai số nguyên l_i, r_i

($1 \leq l_i \leq r_i \leq n$). Nếu phép toán thứ i là loại 2, thì tiếp theo là ba số nguyên l_i, r_i, x_i

($1 \leq l_i \leq r_i \leq n, 1 \leq x_i \leq 10^6$). Các số trên các dòng được phân tách bằng dấu cách đơn.

Đầu ra

Đối với mỗi truy vấn loại 1 in trên một dòng duy nhất tổng của các số trên đoạn cho trước.
In các câu trả lời cho các truy vấn theo thứ tự mà các truy vấn xuất hiện trong đầu vào.

Vui lòng không sử dụng bộ định dạng `%lld` để đọc hoặc viết số nguyên 64-bit trong C++.
Nên sử dụng luồng `cin`, `cout`, hoặc bộ định dạng `%I64d`.

Ví dụ 1

Input	Output
5	26
4 10 3 13 7	22
8	0
1 2 4	34
2 1 3 3	11
1 2 4	
1 3 3	
2 2 5 5	
1 1 5	
2 1 2 10	
1 2 3	

Ví dụ 2

Input	Output
6	38
4 7 4 0 7 3	28
5	
2 2 3 8	
1 1 5	
2 3 5 1	
2 4 5 6	
1 2 3	

Problem N. Quy hoạch động nâng cao

Time Limit 1000 ms

Mem Limit 524288 kB

Cho một dãy n phần tử a_1, a_2, \dots, a_n và một số các truy vấn-k. Ngoài ra còn có một số thao tác cập nhật.

Một thao tác cập nhật là một cặp (i, v) nghĩa là a_i cần được gán giá trị v .

Một truy vấn-k là một bộ ba (i, j, k) ($1 \leq i \leq j \leq n$).

Với mỗi truy vấn-k (i, j, k) , bạn phải trả về số phần tử lớn hơn k nằm trong dãy con a_i, a_{i+1}, \dots, a_j .

Input

- Dòng 1: n ($1 \leq n \leq 30000$).
- Dòng 2: n số a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$).
- Dòng 3: q ($1 \leq q \leq 200000$), số truy vấn và cập nhật.
- q dòng tiếp theo, số đầu tiên trong mỗi dòng là 0 hoặc 1. Số 0 theo sau bởi 2 số i và v ($1 \leq i \leq n, 1 \leq v \leq 10^4$) cho biết một thao tác cập nhật. Số 1 theo sau bởi 3 số nguyên i, j, k ($1 \leq i \leq j \leq n, 1 \leq k \leq 10^4$) cho biết một truy vấn-k.

Output

- Với mỗi truy vấn-k (i, j, k) , in ra số phần tử lớn hơn k trong dãy con a_i, a_{i+1}, \dots, a_j trên một dòng.

Input	Output
5	2
5 1 2 3 4	1
6	2
1 2 4 1	
0 4 10	
1 4 4 4	
0 3 1	
0 1 2	
1 1 5 2	

Problem O. K-query VIII

Time Limit 2000 ms

Mem Limit 262144 kB

Tan claims to be the most handsome guy in the IT class. However, Hieu disagrees and has challenged Tan with a simple problem to see who is truly the most handsome?

The problem is as follows: Given a sequence of n elements and a constant value k . Given q queries as follows:

- 1 $l\ r\ x$: Decrease by an amount x in the range $[l, r]$.
- 2 $l\ r$: Count the number of elements whose value does not exceed k .

Input

The input data consists of:

- The first line: 2 numbers n, k ($1 \leq n \leq 5 \times 10^5, 1 \leq k \leq 10^9$).
- The second line: the sequence a with n elements ($1 \leq a_i \leq 10^9$).
- The third line: the number of queries q ($1 \leq q \leq 5 \times 10^5$).
- The next q lines belong to 1 of the 2 types of queries:
 - 1 $l\ r\ x$: Decrease by an amount x in the range $[l, r]$ ($1 \leq x \leq 10^9$).
 - 2 $l\ r$: Count the number of elements in the range $[l, r]$ whose value does not exceed k .

Output

The output data:

The answer for each query of type 2 on each line.

Scoring

Subtask	Points	Limits
1	20%	$1 \leq n, q \leq 1000$.
2	30%	$1 \leq n, q \leq 10^5$.
3	50%	$1 \leq n, q \leq 5 \times 10^5$.
Input		Output
8 4 3 10 5 8 4 1 2 1 6 2 3 7 2 2 3 1 2 6 4 2 1 5 1 1 4 9 2 1 8		3 0 4 8

Notes

In the first query: 2 3 7, in the range at positions $[3; 7]$ the corresponding numbers are $[5; 8; 4; 1; 2]$ with 3 numbers satisfying $\leq k = 4$ which are 4; 1; 2, so the answer to this query is 3.

In the second query: 2 2 3, in the range at positions $[2; 3]$ the corresponding numbers are $[10; 5]$ with no numbers satisfying $\leq k = 4$, so the answer to this query is 0.

In the third query: 1 2 6 4, we decrease the numbers in the range at positions $[2; 6]$ by 4 units, resulting in the sequence after the operation being $[3; 6; 1; 4; 0; -3; 2; 1]$.

In the fourth query: 2 1 5, in the range at positions $[1; 5]$ the corresponding numbers are $[3; 6; 1; 4; 0]$ with 4 numbers satisfying $\leq k = 4$ which are 3; 1; 4; 0, so the answer to this query is 4.

In the fifth query: 1 1 4 9, we decrease the numbers in the range at positions $[1; 4]$ by 9 units, resulting in the sequence after the operation being $[-6; -3; -8; -5; 0; -3; 2; 1]$.

In the sixth query: 2 1 8, in the range at positions $[1; 8]$ the corresponding numbers are $[-6; -3; -8; -5; 0; -3; 2; 1]$ with 8 numbers satisfying $\leq k = 4$ which are all the numbers in the sequence, so the answer to this query is 8.

Problem P. Thanh gươm diệt quỷ

Time Limit 5000 ms

Mem Limit 524288 kB

Mirko là một người đàn ông rất giản dị. Bạn của Mirko, Darko đã cho anh ta một mảng gồm N số nguyên dương và yêu cầu anh ta trả lời Q truy vấn liên quan đến mảng đã cho.

Mỗi truy vấn bao gồm hai số nguyên, thể hiện vị trí đầu và cuối của một đoạn trong mảng. Câu trả lời cho truy vấn ấy là số lượng các giá trị khác nhau xuất hiện đúng hai lần trong đoạn đã cho.

Input

Dòng đầu tiên chứa số nguyên N và Q ($1 \leq N, Q \leq 500000$).

Dòng thứ hai chứa mảng gồm N số nguyên dương nhỏ hơn 10^9 .

Q dòng tiếp theo mỗi dòng chứa hai số nguyên L và R ($1 \leq L \leq R \leq N$).

Output

Hãy xuất ra Q dòng lần lượt là kết quả của Q truy vấn.

Input	Output
5 1 1 2 1 1 1 1 3	1

Giải thích test ví dụ 1: trong đoạn $[1, 3]$ chỉ có một số duy nhất (1) xuất hiện đúng hai lần.

Input	Output
5 2 1 1 1 1 1 2 4 2 3	0 1
Input	Output
5 2 1 1 2 2 3 1 1 1 5	0 2

Subtask

- 40% số test có $N, Q \leq 5000$
 - 60% số test còn lại không có điều kiện gì thêm
-

Problem Q. Truy vấn trên cây

Time Limit 2000 ms

Mem Limit 524288 kB

Input File stdin

Output File stdout

Valera rất thích các đoạn thẳng. Gần đây anh ấy đã nghĩ ra một bài toán thú vị.

Trục tọa độ Ox có n đoạn thẳng, đoạn thẳng thứ i bắt đầu tại vị trí l_i và kết thúc tại vị trí r_i (chúng ta sẽ gọi đoạn này là $[l_i, r_i]$). Nhiệm vụ của bạn là xử lý m truy vấn, mỗi truy vấn gồm một số cnt_i và một tập hợp cnt_i tọa độ các điểm nằm trên trục Ox . Kết quả của truy vấn là số đoạn thẳng sao cho mỗi đoạn chứa ít nhất một điểm từ truy vấn đó. Đoạn thẳng $[l, r]$ chứa điểm q , nếu $l \leq q \leq r$.

Valera thấy bài toán này quá khó. Vì vậy anh ấy nhờ bạn giúp đỡ. Hãy giúp Valera nhé.

Input

Dòng đầu tiên chứa hai số nguyên n, m ($1 \leq n, m \leq 3 \cdot 10^5$) — số đoạn thẳng trên trục tọa độ và số truy vấn.

Tiếp theo là n dòng mô tả các đoạn thẳng. Dòng thứ i chứa hai số nguyên dương l_i, r_i ($1 \leq l_i \leq r_i \leq 10^6$) — biên giới của đoạn thẳng thứ i .

Tiếp theo là m dòng mô tả các truy vấn, mỗi dòng một truy vấn. Mỗi dòng bắt đầu bằng số nguyên cnt_i ($1 \leq cnt_i \leq 3 \cdot 10^5$) — số điểm trong truy vấn thứ i . Sau đó là cnt_i số nguyên dương khác nhau $p_1, p_2, \dots, p_{cnt_i}$ ($1 \leq p_1 < p_2 < \dots < p_{cnt_i} \leq 10^6$) — tọa độ các điểm trong truy vấn thứ i .

Đảm bảo tổng số điểm trong tất cả các truy vấn không vượt quá $3 \cdot 10^5$.

Output

In ra m số nguyên không âm, trong đó số thứ i là kết quả của truy vấn thứ i .

Examples

Input	Output
3 3 1 3 4 5 6 7 3 1 4 7 2 4 5 1 8	3 1 0

Problem R. Truy vấn xàm xí

Time Limit 3000 ms

Mem Limit 1024 kB

Cho một dãy gồm n số nguyên không âm a_1, a_2, \dots, a_n .

Bạn được yêu cầu trả lời q truy vấn online:

- $1 \ l \ r \ type$: với $type = 1$, sắp xếp a_l, a_{l+1}, \dots, a_r tăng dần. Ngược lại, với $type = 2$, sắp xếp a_l, a_{l+1}, \dots, a_r giảm dần.
- $2 \ l \ r$: Tìm $MEX(a_l, a_{l+1}, \dots, a_r)$.

Chú ý:

- MEX của một dãy số nguyên b gồm (b_1, b_2, \dots, b_m) là số nguyên không âm nhỏ nhất không xuất hiện trong dãy đó.
- Ví dụ: $MEX(0, 1, 2, 4, 5) = 3$, $MEX(1, 1, 5, 100) = 0$.

Input

Vào từ file văn bản `mexquery.inp`:

Dòng đầu tiên gồm số nguyên dương n - độ dài của dãy a ($1 \leq n \leq 3 \cdot 10^5$).

Dòng thứ hai gồm n số nguyên không âm a_1, a_2, \dots, a_n ($0 \leq a_i \leq 30$).

Dòng thứ ba gồm số nguyên dương q - số truy vấn bạn cần trả lời ($1 \leq q \leq 5 \cdot 10^4$).

Trong q dòng cuối cùng, dòng thứ i ($1 \leq i \leq q$) có dạng như sau:

- $1 \ l \ r \ type$ ($1 \leq l, r \leq n, 1 \leq type \leq 2$): Với $type = 1$, sắp xếp a_l, a_{l+1}, \dots, a_r tăng dần. Ngược lại, với $type = 2$, sắp xếp a_l, a_{l+1}, \dots, a_r giảm dần.
- $2 \ l \ r$ ($1 \leq l, r \leq n$): Tìm $MEX(a_l, a_{l+1}, \dots, a_r)$.

Output

In ra file văn bản `mexquery.out`:

- Gồm k dòng tương ứng k truy vấn loại 2, dòng thứ i gồm một số nguyên là kết quả của truy vấn thứ i .

Scoring

Subtask	Điểm	Giới hạn
1	10%	$n \leq 1000, q \leq 1000$, Chỉ bao gồm truy vấn thứ hai.
2	20%	$n \leq 1000, q \leq 1000$.
3	30%	Chỉ bao gồm truy vấn thứ hai.
4	40%	Không có ràng buộc gì thêm.

Input	Output
5 1 3 2 0 4 5 2 1 4 1 1 4 1 2 2 5 1 2 5 2 2 1 3	4 0 1

Problem S. Luồng cực đại

Time Limit 1000 ms

Mem Limit 524288 kB

As a student majoring in Information Technology, Nam often hones his logical thinking and programming skills through competitive programming problems. One interesting problem that Nam is contemplating to solve is as follows:

Given n segments of integers on the number line, where the k^{th} segment ($1 \leq k \leq n$) has its left endpoint at L_k , right endpoint at R_k , and weight of w_k . For two integers a and b , the weight of the pair (a, b) is calculated as the sum of weights of all segments t where $a \leq L_t \leq R_t \leq b$ for $1 \leq t \leq n$. The task is to find the pair of integers (a, b) with the maximum weight.

Requirements: Given n segments of numbers, let S be the weight of the pair of integers (a, b) with the maximum weight, help Nam determine the value of S .

Data

Input from the text file **SSEQ.INP**:

- The first line contains a positive integer n ;
- The k^{th} line ($1 \leq k \leq n$) among the next n lines contains three integers L_k, R_k, w_k describing the k^{th} segment ($1 \leq L_k \leq R_k \leq 10^6; |w_k| \leq 10^6$).

Numbers on the same line are separated by spaces.

Result

Write to the text file **SSEQ.OUT** an integer S which is the maximum weight determined.

- 30% of the test cases, worth 30% of the points, satisfy: $n \leq 200$;
- Another 30% of the test cases, worth 30% of the points, satisfy: $n \leq 2000$;
- 20% of the test cases, worth 20% of the points, satisfy: $R_1 - L_1 = R_2 - L_2 = \dots = R_n - L_n$ and $n \leq 10^5$;
- The remaining 20% of the test cases, worth 20% of the points, satisfy: $n \leq 10^5$.

Example

Input	Output
4 1 2 -5 3 5 6 3 4 -1 4 6 3	8

Explanation

The maximum weight is 8 by selecting the pair (3, 6). The weight of the pair (3, 6) is equal to the sum of weights of three segments: [3, 5], [3, 4], and [4, 6].

Problem T. Alvin and the Chipmunks

Time Limit 1000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Chú Sóc tìm thấy một hàng với n cột. Cột thứ i có chiều cao là h_i mét. Bắt đầu từ một cột i_1 , Chú Sóc muốn nhảy qua các cột i_2, \dots, i_k ($1 \leq i_1 < i_2 < \dots < i_k \leq n$). Từ một cột i , Chú Sóc có thể nhảy lên cột j chỉ nếu $i < j$ và $|h_i - h_j| \geq d$, trong đó $|x|$ là giá trị tuyệt đối của số x .

Bây giờ Chú Sóc đang yêu cầu bạn tìm ra một chuỗi nhảy có độ dài tối đa và in ra nó.

Nhập

Dòng đầu tiên chứa hai số nguyên n và d ($1 \leq n \leq 10^5, 0 \leq d \leq 10^9$).

Dòng thứ hai chứa n số h_1, h_2, \dots, h_n ($1 \leq h_i \leq 10^{15}$).

Đầu ra

Dòng đầu tiên nên chứa một số nguyên k , độ dài tối đa của một chuỗi nhảy.

Dòng thứ hai nên chứa k số nguyên i_1, i_2, \dots, i_k ($1 \leq i_1 < i_2 < \dots < i_k \leq n$), đại diện cho các chỉ số của các cột từ chuỗi nhảy có độ dài tối đa.

Nếu có nhiều hơn một chuỗi nhảy có độ dài tối đa, in ra bất kỳ chuỗi nào.

Ví dụ 1

Input	Output
5 2 1 3 6 7 4	4 1 2 3 5

Ví dụ 2

Input	Output
10 3 2 1 3 6 9 11 7 3 20 18	6 1 4 6 7 8 9

Ghi chú

Trong ví dụ đầu tiên, Chú Sóc chọn các cột 1, 2, 3, 5 với chiều cao là 1, 3, 6, 4. Một chuỗi nhảy khác có độ dài 4 là 1, 2, 4, 5.

Problem U. Tiệc mukbang của ếch

Time Limit 2000 ms

Mem Limit 262144 kB

Có n chú ếch đang ngồi trên trục tọa độ Ox . Với mỗi chú ếch, ta biết hai giá trị x_i và t_i lần lượt là vị trí ngồi và độ dài lưỡi ban đầu của chú ếch thứ i . m con muỗi sẽ lần lượt hạ cánh xuống trục Ox . Với mỗi con muỗi, ta biết hai giá trị p_j và b_j lần lượt là vị trí hạ cánh và độ lớn của con muỗi thứ j . Các chú ếch và muỗi có thể được biểu diễn bằng các điểm trên trục tọa độ.

Một chú ếch có thể ăn một con muỗi nếu như con muỗi ở cùng vị trí với chú ếch hoặc ở phía bên phải với khoảng cách giữa chú ếch và con muỗi không lớn hơn độ dài lưỡi của chú ếch đó.

Nếu tại cùng một thời điểm, nhiều chú ếch có thể ăn được một con muỗi, thì chú ếch ở vị trí trái nhất sẽ ăn nó (chú ếch có giá trị x_i bé nhất). Sau khi ăn một con muỗi, độ dài lưỡi của một chú ếch sẽ tăng đúng bằng độ lớn của con muỗi đó. Sau đó, chú ếch sẽ ăn những con muỗi khác nếu như có thể (chưa bị ăn trong các lượt trước đó).

Với mỗi chú ếch, hãy in ra hai giá trị – số lượng con muỗi đã ăn và độ dài lưỡi sau khi tất cả con muỗi đã hạ cánh và không còn con muỗi nào trên trục tọa độ có thể bị ăn.

Những con muỗi sẽ hạ cánh xuống trục tọa theo thứ tự của input. Mỗi con muỗi sẽ chỉ hạ cánh sau khi tất cả con muỗi đã hạ cánh trước đó đã bị ăn hoặc không thể bị ăn.

Input

Dòng đầu tiên gồm hai số nguyên n, m ($1 \leq n, m \leq 2 \times 10^5$) lần lượt là số chú ếch và số con muỗi.

n dòng tiếp theo, mỗi dòng gồm hai số nguyên x_i, t_i ($0 \leq x_i, t_i \leq 10^9$) lần lượt là vị trí và độ dài lưỡi ban đầu của chú ếch thứ i . Input đảm bảo không có hai chú ếch nào sẽ ngồi ở cùng một vị trí.

m dòng tiếp theo, mỗi dòng gồm hai số nguyên p_j, b_j ($0 \leq p_j, b_j \leq 10^9$) lần lượt là vị trí và độ lớn của con muỗi thứ j .

Output

In ra n dòng, dòng thứ i gồm hai số nguyên lần lượt là số con muỗi mà chú ếch thứ i ăn được và độ dài cuối cùng của lưỡi của của chú ếch thứ i .

Sample 1

Input	Output
4 6 10 2 15 0 6 1 0 1 110 10 1 1 6 0 15 10 14 100 12 2	3 114 1 10 1 1 1 2

Sample 2

Input	Output
1 2 10 2 20 2 12 1	1 3

Problem V. Làng xì trum

Time Limit 1000 ms

Mem Limit 262144 kB

Vương quốc Valanro có N công trình, mỗi công trình có mã số là a_i ($1 \leq i \leq N$), vì muốn tối ưu hóa việc quản lý dân cư, quốc vương KaJo quyết định gộp các công trình thành M đại công trình.

- Một đại công trình sẽ được dựng lên bằng các công trình liên kế nhau (Đại công trình bao gồm tối thiểu **một** công trình).
- Mã số của đại công trình là ước chung lớn nhất của các mã số của các công trình trong đại công trình đó.

Xây dựng một đại công trình sẽ tốn khá nhiều công sức, vì thế quốc vương đã thỏa thuận với đội thi công như sau:

- Một đại công trình có mã số là K sẽ tốn chi phí là X để xây dựng.
- Một đại công trình có mã số **khác** K sẽ tốn chi phí là Y để xây dựng.

Ví dụ: Cho 5 công trình có các mã số là $[8, 4, 6, 10, 14]$ cần dựng lên 3 đại công trình với K, X, Y lần lượt là 2, 2, 3. Dựng các đại công trình có mã số là $[8, 2, 2]$ bằng cách gộp các công trình $[8], [4, 6], [10, 14]$ với chi phí là $3 + 2 + 2 = 7$.

Hãy giúp quốc vương KaJo tìm cách dựng các đại công trình có chi phí **thấp nhất**.

Input Format

- Dòng thứ nhất gồm 2 số nguyên dương N, M ($N \leq 10^4, M \leq \min(20, N)$).
- Dòng thứ hai gồm 3 số nguyên dương K, X, Y ($K, X, Y \leq 10^9$).
- Dòng thứ ba gồm N số nguyên dương a_i ($a_i \leq 10^9$).

Output Format

- In ra chi phí thấp nhất để vương quốc thực hiện dự án.

Subtasks

- Có 20% số lượng test thỏa mãn điều kiện: $N \leq 10$.
- Có 30% số lượng test khác thỏa mãn điều kiện: $N \leq 2000$.
- Có 50% số lượng test còn lại thỏa mãn điều kiện: $N \leq 10^4$.

Input	Output
5 3 2 2 3 8 4 6 10 14	7
Input	Output
8 4 3 3 4 9 6 12 6 6 9 9 12	13

Problem W. Số may mắn

Time Limit 4000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Petya yêu thích các số may mắn. Mọi người đều biết rằng các số may mắn là các số nguyên dương mà biểu diễn thập phân của chúng chỉ chứa các chữ số may mắn 4 và 7. Ví dụ, các số 47, 744, 4 là số may mắn và các số 5, 17, 467 không phải.

Petya có một mảng gồm n số. Anh ấy muốn thực hiện m phép toán của hai loại sau:

- add $l\ r\ d$ — thêm một số nguyên d vào tất cả các phần tử có chỉ số thuộc khoảng từ l đến r , bao gồm cả ($1 \leq l \leq r \leq n, 1 \leq d \leq 10^4$);
- count $l\ r$ — tìm và in ra màn hình có bao nhiêu số may mắn trong các phần tử có chỉ số thuộc khoảng từ l đến r bao gồm cả ($1 \leq l \leq r \leq n$). Mỗi số may mắn sẽ được đếm mỗi lần xuất hiện trong khoảng.

Petya có một danh sách tất cả các phép toán. Các phép toán được thiết lập sao cho sau tất cả các phép cộng, mảng sẽ không có số nào vượt quá 10^4 . Hãy giúp Petya viết một chương trình để thực hiện các phép toán này.

Nhập

Dòng đầu tiên chứa hai số nguyên n và m ($1 \leq n, m \leq 10^5$) — số lượng số trong mảng và số lượng phép toán tương ứng. Dòng thứ hai chứa n số nguyên dương, không vượt quá 10^4 — đó là các số trong mảng. Tiếp theo m dòng chứa các phép toán, mỗi dòng một phép. Chúng tương ứng với mô tả trong bài toán.

Đảm bảo rằng sau khi thực hiện tất cả các phép toán, mỗi số trong mảng sẽ không vượt quá 10^4 .

Xuất

Đối với mỗi phép toán loại thứ hai, in ra một số duy nhất trên một dòng — số lượng số

may mắn trong khoảng tương ứng.

Ví dụ

Input	Output
3 6 2 3 4 count 1 3 count 1 2 add 1 3 2 count 1 3 add 2 3 3 count 1 3	1 0 1 1

Input	Output
4 5 4 4 4 4 count 1 4 add 1 4 3 count 1 4 add 2 3 40 count 1 4	4 4 4

Ghi chú

Trong ví dụ đầu tiên sau phép cộng đầu tiên, mảng sẽ trở thành như sau:

4 5 6

Sau phép cộng thứ hai:

4 8 9

Ví dụ thứ hai sau phép cộng đầu tiên:

7 7 7 7

Sau phép cộng thứ hai:

7 47 47 7

Problem X. Dragon Ball

Time Limit 2000 ms

Mem Limit 524288 kB

Input File stdin

Output File stdout

Drazil là một con khi. Nó sống trong một công viên hình tròn. Có n cây xung quanh công viên. Khoảng cách giữa cây thứ i và cây thứ $(i + 1)$ là d_i , khoảng cách giữa cây thứ n và cây đầu tiên là d_n . Chiều cao của cây thứ i là h_i .

Drazil bắt đầu mỗi ngày với *cuộc chạy buổi sáng*. Cuộc chạy buổi sáng bao gồm các bước sau:

- Drazil chọn hai cây khác nhau
- Nó bắt đầu bằng cách leo lên cây đầu tiên
- Sau đó nó leo xuống cây đầu tiên, chạy quanh công viên (theo một trong hai hướng có thể) đến cây thứ hai, và leo lên cây đó
- Cuối cùng nó leo xuống cây thứ hai.

Nhưng luôn có trẻ em chơi quanh một số cây liên tiếp. Drazil không thể chịu đựng được trẻ em, vì vậy nó không thể chọn những cây gần trẻ em. Nó thậm chí không thể ở gần những cây đó.

Nếu hai cây mà Drazil chọn là cây thứ x và cây thứ y , chúng ta có thể ước lượng năng lượng mà *cuộc chạy buổi sáng* tiêu tốn của nó là $2(h_x + h_y) + \text{dist}(x, y)$. Vì có trẻ em trên chính một trong hai cung nối x và y , khoảng cách $\text{dist}(x, y)$ giữa cây x và cây y được xác định duy nhất.

Bây giờ, bạn biết rằng vào ngày i trẻ em chơi giữa cây thứ a_i và cây thứ b_i . Nói một cách chính thức, nếu $a_i \leq b_i$, trẻ em chơi quanh các cây có chỉ số trong khoảng $[a_i, b_i]$, nếu không thì chúng chơi quanh các cây có chỉ số từ $[a_i, n] \cup [1, b_i]$.

Xin hãy giúp Drazil xác định hai cây mà nó nên chọn để tiêu tốn nhiều năng lượng nhất (vì nó muốn trở thành một con khi khỏe mạnh và đẹp trai) và báo cáo số năng lượng thu được cho mỗi ngày.

Input

Dòng đầu tiên chứa hai số nguyên n và m ($3 \leq n \leq 10^5$, $1 \leq m \leq 10^5$), biểu thị số lượng cây và số lượng ngày, tương ứng.

Dòng thứ hai chứa n số nguyên d_1, d_2, \dots, d_n ($1 \leq d_i \leq 10^9$), khoảng cách giữa các cây liên tiếp.

Dòng thứ ba chứa n số nguyên h_1, h_2, \dots, h_n ($1 \leq h_i \leq 10^9$), chiều cao của các cây.

Mỗi dòng tiếp theo m chứa hai số nguyên a_i và b_i ($1 \leq a_i, b_i \leq n$) mô tả mỗi ngày mới. Luôn có ít nhất hai cây khác nhau mà Drazil có thể chọn không bị ảnh hưởng bởi trẻ em.

Output

Đối với mỗi ngày, in câu trả lời trên một dòng riêng biệt.

Examples

Input	Output
5 3 2 2 2 2 2 3 5 2 1 4 1 3 2 2 4 5	12 16 18

Input	Output
3 3 5 1 4 5 1 4 3 3 2 2 1 1	17 22 11

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐỀ THI CHÍNH THỨC

KỲ THI CHỌN HỌC SINH GIỎI QUỐC GIA THPT
NĂM HỌC 2020-2021

Môn: TIN HỌC

Thời gian: 180 phút (không kể thời gian giao đề)

Ngày thi thứ nhất: 25/12/2020

(Đề thi có 04 trang, gồm 03 bài)

TỔNG QUAN NGÀY THI THỨ NHẤT

	Tên bài	File chương trình	File dữ liệu vào	File kết quả
Bài 1	Tặng quà	NOEL.*	NOEL.INP	NOEL.OUT
Bài 2	Mạng truyền thông	COMNET.*	COMNET.INP	COMNET.OUT
Bài 3	Phép toán OR	OR.*	OR.INP	OR.OUT

Dấu * được thay thế bởi PAS hoặc CPP của ngôn ngữ lập trình được sử dụng tương ứng là Pascal hoặc C++.

Hãy lập trình giải các bài toán sau:

Bài 1. Tặng quà (7 điểm)

Noel sắp tới, Ông Già Tuyết đã chuẩn bị $2n$ món quà dành cho các bạn nhỏ. Các món quà có màu sắc đôi một khác nhau và có mã màu từ 1 đến $2n$. Khi cho các món quà vào túi, Ông đã đưa các món quà vào theo một thứ tự mà nếu lấy ra, các món quà sẽ có mã màu lần lượt là c_1, c_2, \dots, c_{2n} (dãy c_1, c_2, \dots, c_{2n} là một hoán vị của $1, 2, \dots, 2n$).

Ông Già Tuyết dự định tặng quà cho m ($m \leq n$) bạn nhỏ, mỗi bạn sẽ được nhận hai món quà sau hai lượt tặng. Các bạn nhỏ đứng thành một hàng và Ông sẽ đi từ đầu hàng đến cuối hàng để lần lượt tặng quà cho từng bạn. Khi đứng trước một bạn nhỏ để tặng quà, Ông lần lượt lấy từng món quà ra cho tới khi lựa chọn được một món quà phù hợp và tặng bạn nhỏ, các món quà không được lựa chọn sẽ được cất đi và không được dùng để tặng quà. Khi bạn nhỏ thứ m ở cuối hàng đã được nhận quà, Ông sẽ di chuyển về đầu hàng để tặng quà lượt thứ hai tương tự như lượt thứ nhất.

Ông được biết, các bạn nhỏ luôn mong muốn nhận được hai món quà mà chênh lệch mã màu của hai món quà đó không vượt quá d . Với mong muốn mang lại nhiều niềm vui cho các bạn nhỏ, Ông quyết định việc tặng quà sẽ phải bảo đảm tất cả các bạn nhỏ đều nhận được hai món quà mà chênh lệch mã màu không vượt quá d .

Một cách hình thức, gọi m là số lượng bạn nhỏ được tặng quà, Ông cần chọn ra dãy $2m$ chỉ số $1 \leq i_1 < i_2 < \dots < i_m < i_{m+1} < \dots < i_{2m} \leq 2n$ sao cho $|c_{i_k} - c_{i_{m+k}}| \leq d$ với mọi $1 \leq k \leq m$.

Ông Già Tuyết biết rằng, có thể không tồn tại cách chọn được $2m$ chỉ số thỏa mãn, điều đó cũng có nghĩa là không thể tặng quà như mong muốn cho cả m bạn nhỏ. Do đó, với một số nguyên dương d và thứ tự các món quà lấy ra có mã màu lần lượt là c_1, c_2, \dots, c_{2n} , Ông muốn tính số lượng nhiều nhất các bạn nhỏ mà Ông có thể tặng quà.

Yêu cầu: Hãy giúp Ông Già Tuyết tính số lượng nhiều nhất các bạn nhỏ mà Ông có thể tặng quà đáp ứng điều kiện nêu trên.

Dữ liệu: Vào từ file văn bản NOEL.INP:

- Dòng thứ nhất chứa hai số nguyên dương n và d ($d \leq 5$);
- Dòng thứ hai chứa $2n$ số nguyên dương c_1, c_2, \dots, c_{2n} là mã màu của các món quà lần lượt được lấy ra.

Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả: Ghi ra file văn bản NOEL.OUT một số nguyên duy nhất là số lượng nhiều nhất các bạn nhỏ mà Ông Già Tuyết có thể tặng quà.

Ràng buộc:

- Có 40% số test ứng với 40% số điểm của bài thỏa mãn: $n \leq 10$;
- 40% số test khác ứng với 40% số điểm của bài thỏa mãn: $n \leq 100$;
- 20% số test còn lại ứng với 20% số điểm của bài thỏa mãn: $n \leq 1000$.

Ví dụ:

NOEL.INP	NOEL.OUT	Giải thích
<pre> 3 1 1 5 6 3 4 2 </pre>	2	<p>Ông Già Tuyết có thể tặng tối đa cho 2 bạn nhỏ.</p> <ul style="list-style-type: none"> - Lượt thứ nhất, món quà có mã màu 5 tặng bạn thứ nhất, món quà có mã màu 3 tặng bạn thứ hai. - Lượt thứ hai, món quà có mã màu 4 tặng bạn thứ nhất và món quà có mã màu 2 tặng bạn thứ hai.

Bài 2. Mạng truyền thông (7 điểm)

Một ngân hàng có n chi nhánh, mỗi chi nhánh có một máy chủ, các máy chủ ở các chi nhánh được đánh số từ 1 đến n . Nhằm bảo đảm việc truyền thông giữa các chi nhánh, ngân hàng đã thuê $n - 1$ kênh truyền tin trực tiếp từ một nhà cung cấp dịch vụ mạng để kết nối n máy chủ thành một mạng máy tính và bảo đảm từ máy chủ của một chi nhánh bất kì có thể truyền tin đến tất cả các máy chủ của các chi nhánh còn lại theo kênh truyền tin trực tiếp giữa chúng hoặc thông qua đường truyền đi qua một số máy chủ của các chi nhánh nào đó.

Trong thời gian tới, ngân hàng muốn lựa chọn k máy chủ trong n máy chủ để cài đặt phần mềm kiểm soát. Phần mềm khi hoạt động sẽ làm tăng lưu lượng truyền trên các kênh giữa k máy chủ này. Với mỗi phương án chọn k máy chủ để cài đặt phần mềm, trong số $n - 1$ kênh truyền tin, nhà cung cấp dịch vụ mạng đã xác định một số ít nhất các kênh đủ để kết nối k máy chủ này và khuyến cáo ngân hàng cần phải nâng cấp các kênh đó. Vì lý do kỹ thuật cũng như kinh phí, ngân hàng muốn lựa chọn k máy chủ để cài đặt phần mềm mà số lượng kênh ít nhất cần nâng cấp có giá trị nằm trong đoạn $[a, b]$. Cụ thể, với một cách chọn k máy chủ, gọi s là số kênh ít nhất cần chọn ra trong $n - 1$ kênh truyền tin nhằm bảo đảm liên thông giữa k máy chủ được chọn ngay cả khi các kênh còn lại bị đứt kết nối, s kênh này sẽ được khuyến cáo nâng cấp, khi đó, cách chọn k máy chủ thỏa mãn yêu cầu của ngân hàng nếu $a \leq s \leq b$.

Yêu cầu: Cho $n - 1$ kênh truyền tin và các giá trị k, a, b , hãy đếm số lượng cách chọn k máy chủ để cài đặt phần mềm mà số lượng kênh ít nhất cần nâng cấp nằm trong đoạn $[a, b]$.

Dữ liệu: Vào từ file văn bản COMNET.INP:

- Dòng thứ nhất chứa bốn số nguyên dương n, k, a, b ($k < n; 1 < a \leq b < n$);
- Dòng thứ i trong số $n - 1$ dòng tiếp theo chứa hai số nguyên dương u_i, v_i cho biết có kênh truyền tin trực tiếp giữa hai máy chủ u_i, v_i ($1 \leq u_i, v_i \leq n; u_i \neq v_i$).


Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả: Ghi ra file văn bản COMNET.OUT một số nguyên duy nhất là số cách chọn k máy chủ để cài đặt phần mềm thỏa mãn yêu cầu của ngân hàng.

Ràng buộc:

- Có 20% số test ứng với 20% số điểm của bài thỏa mãn: $n \leq 100$ và $k = 2$;
- 30% số test khác ứng với 30% số điểm của bài thỏa mãn: $n \leq 100$ và $k = 3$;
- 20% số test khác ứng với 20% số điểm của bài thỏa mãn: $n \leq 100$ và $k = 4$;
- 20% số test khác ứng với 20% số điểm của bài thỏa mãn: $n \leq 1000$ và $k = 3$;
- 10% số test còn lại ứng với 10% số điểm của bài thỏa mãn: $n \leq 1000$ và $k = 4$.

Ví dụ:

COMNET.INP	COMNET.OUT	Giải thích
<pre> 6 3 2 3 1 2 2 3 3 4 4 5 3 6 </pre>	14	<p>Có 5 cách chọn 3 máy chủ trong 6 máy chủ mà số kênh cần nâng cấp bằng 2 là: $(1, 2, 3); (2, 3, 4); (2, 3, 6); (3, 4, 5); (3, 4, 6)$.</p> <p>Có 9 cách chọn 3 máy chủ trong 6 máy chủ mà số kênh cần nâng cấp bằng 3 là: $(1, 2, 4); (1, 2, 6); (1, 3, 4); (1, 3, 6); (2, 3, 5); (2, 4, 5); (2, 4, 6); (3, 5, 6); (4, 5, 6)$.</p> <p>Như vậy, có tất cả 14 cách chọn 3 máy chủ trong 6 máy chủ thỏa mãn yêu cầu của ngân hàng.</p> 

Bài 3. Phép toán OR (6 điểm)

Nội dung chính trong tiết Tin học trên lớp của Nam ngày hôm nay là về phép toán OR. Phép toán OR (có kí hiệu là $|$) được định nghĩa như sau: Kết quả của phép toán OR giữa hai số nguyên không âm x và y là một số nguyên không âm z trong đó bit thứ i trong biểu diễn nhị phân của z sẽ là 0 khi và chỉ khi bit thứ i trong biểu diễn nhị phân của x và y đồng thời bằng 0, ngược lại bit thứ i trong biểu diễn nhị phân của z sẽ là 1.

Ví dụ, $x = 12$ có biểu diễn nhị phân là 1100, $y = 5$ có biểu diễn nhị phân là 0101, khi đó $x | y$ có biểu diễn nhị phân là 1101, tức giá trị 13 trong hệ cơ số thập phân.

Phép toán OR có tính chất giao hoán và kết hợp: $x | y = y | x$ và $x | (y | z) = (x | y) | z$.

Để ôn tập nội dung đã học, thầy giáo ra cho cả lớp bài tập về nhà như sau: Cho n số nguyên không âm a_1, a_2, \dots, a_n và ba số nguyên k, L, R . Hãy đếm xem có bao nhiêu bộ k chỉ số (i_1, i_2, \dots, i_k) thỏa mãn đồng thời hai điều kiện:

- $1 \leq i_1 < i_2 < \dots < i_k \leq n$;
- Đặt $v = a_{i_1} | a_{i_2} | \dots | a_{i_k}$ thì $L \leq v \leq R$ và v chia hết cho 3.

Gọi S là đáp số đúng của bài tập thầy giáo ra, vì S có thể có giá trị rất lớn và để học sinh tập trung vào vấn đề chính của bài toán nên thầy giáo chỉ yêu cầu học sinh đưa ra được phần dư của S khi chia cho $10^9 + 7$.

Yêu cầu: Hãy giúp Nam xác định phần dư của S khi chia cho $10^9 + 7$.

Dữ liệu: Vào từ file văn bản OR.INP:

- Dòng đầu tiên chứa bốn số nguyên n, k, L, R ($1 \leq k \leq n \leq 10^6$; $0 \leq L \leq R \leq 10^6$);
- Dòng tiếp theo chứa n số nguyên không âm a_1, a_2, \dots, a_n ($a_i \leq 10^6$);

Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả: Ghi ra file văn bản OR.OUT một số nguyên duy nhất là phần dư của S khi chia cho $10^9 + 7$.

Ràng buộc:

- Có 20% số test ứng với 20% số điểm của bài thỏa mãn: $n \leq 20$ và $a_i \leq 200$ ($1 \leq i \leq n$);
- 20% số test khác ứng với 20% số điểm của bài thỏa mãn: $n \leq 200$ và $a_i \leq 200$ ($1 \leq i \leq n$);
- 20% số test khác ứng với 20% số điểm của bài thỏa mãn: $L = R$ và a_i ($1 \leq i \leq n$) là lũy thừa của 2;
- 20% số test khác ứng với 20% số điểm của bài thỏa mãn: $L = R$;
- 20% số test còn lại ứng với 20% số điểm của bài không có ràng buộc gì thêm.

Ví dụ:

OR.INP	OR.OUT	Giải thích
5 2 1 7 1 2 5 6 4	4	Có tất cả 4 cách chọn 2 số trong 5 số thỏa mãn yêu cầu: $a_1 a_2 = 1 2 = 3$ $a_2 a_4 = 2 6 = 6$ $a_2 a_5 = 2 4 = 6$ $a_4 a_5 = 6 4 = 6$

Hết

- Thí sinh không được sử dụng tài liệu.
- Giám thị không giải thích gì thêm.

Problem Z. Cây DFS phân đoạn

Time Limit 1120 ms

Mem Limit 524288 kB

Đọc đề gốc [tại đây](#).

Thành phố nơi Alice sinh sống có n tòa nhà. Trên bản thiết kế, mặt bằng nền mỗi tòa nhà đều là hình chữ nhật có các cạnh song song với trục tọa độ, với các đỉnh đều có tọa độ nguyên trên hệ trục tọa độ. Hai tòa nhà gọi là **liền kề** nếu có hai đường biên của hai hình chữ nhật mặt nền tương ứng hai tòa nhà đó có ít nhất một điểm chung. Để thuận tiện đi lại giữa các tòa nhà, người ta xây dựng một lối đi tắt hai chiều giữa mỗi cặp tòa nhà **liền kề**.

Alice rất thích thú với cách thiết kế các tòa nhà của thành phố và thường xuyên đi lại giữa các tòa nhà thông qua các lối đi tắt này. Sau một thời gian khám phá, Alice nhận thấy có một số lối đi tắt là độc đạo. Một lối đi tắt giữa hai tòa nhà u và v gọi là độc đạo nếu như sau khi đi từ u sang v qua lối đi này thì không có cách đi nào quay trở lại u thông qua các lối đi tắt khác ngoài lối đi tắt từ v về u .

Với mỗi cặp tòa nhà (u, v) có lối đi tắt là độc đạo, Alice tiếp tục khám phá nếu như đóng cửa lối đi độc đạo này thì xuất phát từ u Alice có thể tham quan được tối đa d_u tòa nhà, và nếu xuất phát từ v Alice có thể tham quan được tối đa d_v tòa nhà. Khoảng chênh lệch giữa d_u và d_v là giá trị $|d_u - d_v|$.

Yêu cầu: Biết tọa độ các đỉnh hình chữ nhật mặt nền tương ứng của các tòa nhà, hãy giúp Alice tìm cặp tòa nhà (u, v) có lối đi tắt độc đạo sao cho khoảng chênh lệch giữa d_u và d_v là nhỏ nhất.

Input

- Dòng thứ nhất chứa một số nguyên dương n là số lượng tòa nhà cao tầng;
- Dòng thứ i trong số n dòng tiếp theo chứa 4 số nguyên không âm x_i, y_i, p_i, q_i với (x_i, y_i) là tọa độ của đỉnh trái trên và (p_i, q_i) là tọa độ của đỉnh phải dưới của hình chữ nhật biểu thị mặt nền tòa nhà thứ i trên bản đồ quy hoạch. Dữ liệu đảm bảo $x_i < p_i$ và $y_i > q_i$ và hai hình chữ nhật bất kỳ có thể tiếp xúc nhưng không giao nhau.

Các số trên cùng một dòng cách nhau bởi dấu cách.

Output

Ghi ra một số nguyên duy nhất là khoảng chênh lệch nhỏ nhất tìm được. Nếu không tồn tại lối đi tắt độc đạo thì in ra số -1 .

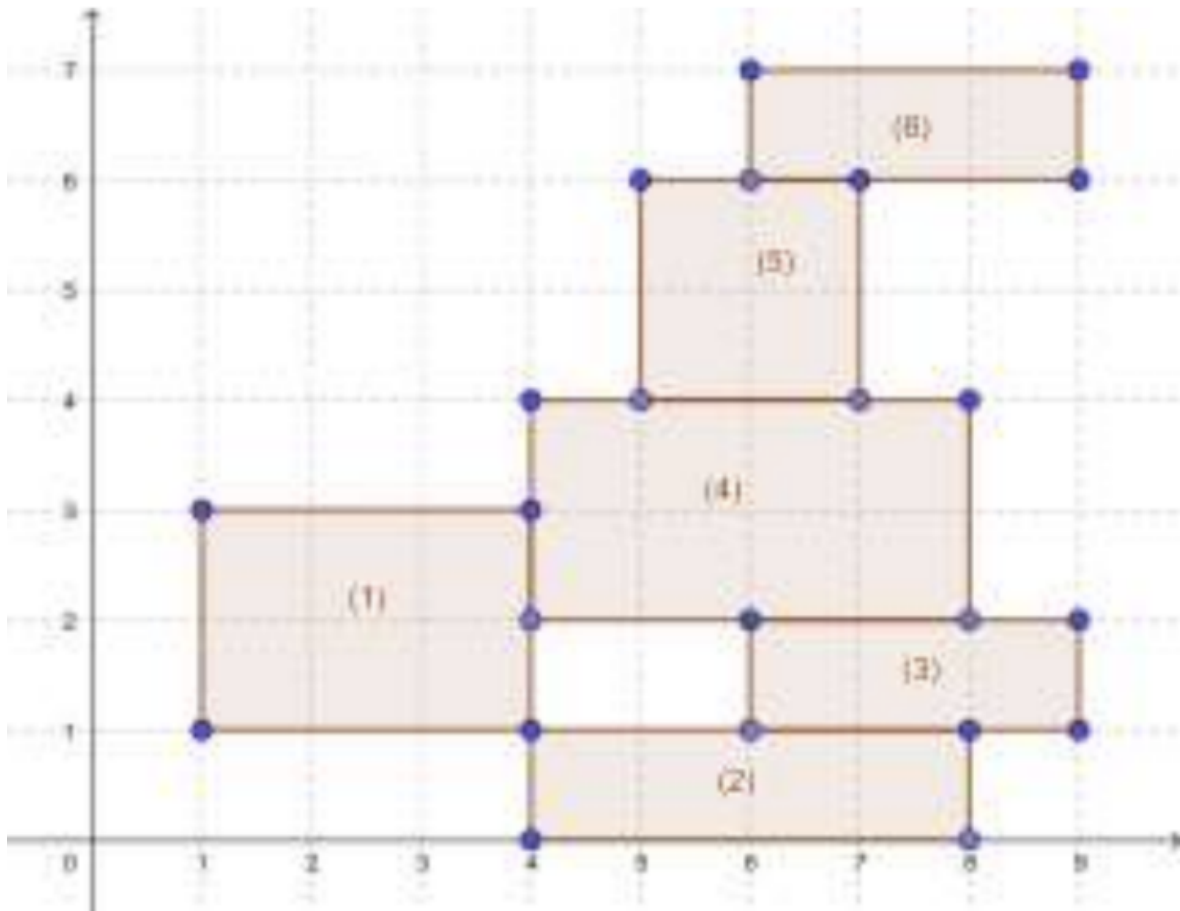
Giới hạn

- Có 30% số test ứng với 30% số điểm của bài thỏa mãn điều kiện: $n \leq 10^3$, các tọa độ không vượt quá 10^9 ;
- 30% số test khác ứng với 30% số điểm của bài thỏa mãn điều kiện: $n \leq 10^5$, các tọa độ không vượt quá 10^3 ;
- 40% số test còn lại ứng với 40% số điểm của bài thỏa mãn điều kiện: $n \leq 10^5$, các tọa độ không vượt quá 10^9 .

Input	Output
6 1 3 4 1 4 1 8 0 6 2 9 1 4 4 8 2 5 6 7 4 6 7 9 6	2

Note

Có 2 lối đi độc đạo là lối đi giữa cặp tòa nhà $(4, 5)$ và $(5, 6)$. Nếu đóng cửa lối đi độc đạo qua $(4, 5)$, ta có $d_4 = 4$ và $d_5 = 2$, do đó $|d_4 - d_5| = 2$. Nếu đóng cửa lối đi độc đạo qua $(5, 6)$, ta có $d_5 = 5$ và $d_6 = 1$, do đó $|d_5 - d_6| = 4$.



Problem AA. Hoán vị - Tổ hợp - Chinh hợp

Time Limit 1000 ms

Mem Limit 524288 kB

Bạn được cho một chuỗi s có độ dài n , bao gồm k chữ cái tiếng Anh viết thường đầu tiên.

Chúng tôi định nghĩa một c -lặp của một chuỗi nào đó q là một chuỗi, bao gồm c bản sao của chuỗi q . Ví dụ, chuỗi "acbacbacbacb" là một 4-lặp của chuỗi "acb".

Giả sử rằng chuỗi a chứa chuỗi b như một chuỗi con, nếu chuỗi b có thể được lấy từ a bằng cách xóa một số ký hiệu.

Để p là một chuỗi đại diện cho một hoán vị nào đó của k chữ cái tiếng Anh viết thường đầu tiên. Chúng tôi định nghĩa hàm $d(p)$ là số nguyên nhỏ nhất sao cho một $d(p)$ -lặp của chuỗi p chứa chuỗi s như một chuỗi con.

Có m thao tác thuộc một trong hai loại có thể được áp dụng cho chuỗi s :

1. Thay thế tất cả các ký tự tại các vị trí từ l_i đến r_i bằng một ký tự c_i .
2. Đối với p đã cho, đó là một hoán vị của k chữ cái tiếng Anh viết thường đầu tiên, tìm giá trị của hàm $d(p)$.

Tất cả các thao tác được thực hiện tuần tự, theo thứ tự chúng xuất hiện trong đầu vào. Nhiệm vụ của bạn là xác định giá trị của hàm $d(p)$ cho tất cả các thao tác loại hai.

Input

Dòng đầu tiên chứa ba số nguyên dương n , m và k

$(1 \leq n \leq 200\,000, 1 \leq m \leq 20\,000, 1 \leq k \leq 10)$ — độ dài của chuỗi s , số lượng thao tác và kích thước của bảng chữ cái tương ứng. Dòng thứ hai chứa chuỗi s chính nó.

Mỗi dòng tiếp theo m chứa một mô tả về một thao tác nào đó:

1. Thao tác loại một bắt đầu bằng 1 theo sau là một bộ ba l_i , r_i và c_i , biểu thị việc thay thế tất cả các ký tự tại các vị trí từ l_i đến r_i bằng ký tự c_i ($1 \leq l_i \leq r_i \leq n$, c_i là một trong k chữ cái tiếng Anh viết thường đầu tiên).
2. Thao tác loại hai bắt đầu bằng 2 theo sau là một hoán vị của k chữ cái tiếng Anh viết thường đầu tiên.

Output

Đối với mỗi truy vấn của loại hai, giá trị của hàm $d(p)$.

Examples

Input	Output
7 4 3 abacaba 1 3 5 b 2 abc 1 4 4 c 2 cba	6 5

Note

Sau thao tác đầu tiên, chuỗi s sẽ là `abbbbba`.

Trong thao tác thứ hai, câu trả lời là 6-lặp của `abc`: `ABcaBcaBcaBcaBcAbc`.

Sau thao tác thứ ba, chuỗi s sẽ là `abbcbba`.

Trong thao tác thứ tư, câu trả lời là 5-lặp của `cba`: `cbAcBacBaCBacBA`.

Các chữ cái viết hoa có nghĩa là sự xuất hiện của các ký hiệu từ chuỗi s .

Problem AB. Kung Fu Panda

Time Limit 1000 ms

Mem Limit 262144 kB

Bạn được cho một dãy số A_1, A_2, \dots, A_n và m tập hợp S_1, S_2, \dots, S_m các chỉ số của mảng này. Gọi $S_k = \{S_{k,i}\}, 1 \leq i \leq |S_k|$. Nói cách khác, $S_{k,i}$ là một phần tử bất kỳ từ tập hợp S_k .

Trong bài này bạn sẽ phải trả lời q truy vấn có 2 dạng sau:

- $? k$: Tính tổng $\sum_{i=1}^{|S_k|} A_{S_{k,i}}$, hay tổng các phần tử có vị trí thuộc tập S_k của dãy A .
- $+ k x$: Cộng x vào các phần tử của dãy A có chỉ số trong tập S_k . Phần tử $A_{S_{k,i}}$ được thay bằng $A_{S_{k,i}} + x$ với mọi $i \in [1, |S_k|]$.

Với mỗi truy vấn loại đầu tiên hãy in tổng đã tính.

Input

Dòng đầu tiên gồm 3 số n, m, q ($1 \leq n, m, q \leq 10^5$). Dòng thứ hai gồm n phần tử A_1, A_2, \dots, A_n ($|A_i| \leq 10^8$), các phần tử của dãy A .

m dòng tiếp theo, dòng thứ k gồm một số nguyên dương ở đầu cho biết số lượng phần tử của tập S_k , theo sau bởi $|S_k|$ số nguyên dương phân biệt $S_{k,1}, S_{k,2}, \dots, S_{k,|S_k|}$ ($1 \leq S_{k,i} \leq n$).

q dòng tiếp theo, mỗi dòng có dạng $? k$ hoặc $+ k x$. Với mọi truy vấn có $1 \leq k \leq m, |x| \leq 10^8$. Các truy vấn được cho theo thứ tự chúng cần được trả lời.

Đề đảm bảo tổng của kích thước mọi tập S_k không quá 10^5 .

Output

Sau mỗi truy vấn dạng đầu tiên hãy in tổng đã tính trên một dòng.

Input	Output
5 3 5 5 -5 5 1 -4 2 1 2 4 2 1 4 5 2 2 5 ? 2 + 3 4 ? 1 + 2 1 ? 2	-3 4 9

Notes

Ở truy vấn đầu tiên, tập S_k là S_2 có 4 phần tử $\{2, 1, 4, 5\}$. Tổng của 4 phần tử có vị trí trong tập S_2 là $A_2 + A_1 + A_4 + A_5 = -5 + 5 + 1 + (-4) = -3$.

Problem AC. Bài khó

Time Limit 2000 ms

Mem Limit 262144 kB

Nam vừa đoạt giải quán quân trong một kỳ thi lập trình danh giá. Ban tổ chức trao thưởng thông qua một trò chơi như sau. Có n thẻ xếp trên một hàng, được đánh số thứ tự từ 1 đến n từ trái sang phải, trên thẻ thứ i ($1 \leq i \leq n$) có hai thông tin:

- Nhãn c_i là một xâu kí tự gồm các kí tự chữ cái la tinh in hoa (từ **A** đến **Z**);
- Số may mắn p_i , là một số nguyên dương.

Ban tổ chức cũng công bố m cặp số may mắn và cho phép Nam thực hiện một hoặc nhiều bước chọn các thẻ. Ban đầu, Nam chọn một thẻ bất kì, điểm nhận được là số may mắn trên thẻ đó. Mỗi bước tiếp theo, Nam thực hiện theo một trong hai cách sau:

1. Trong các thẻ ở phía bên phải thẻ hiện tại, chọn một thẻ có nhãn là một xâu mà có thể nhận được bằng cách xóa từ xâu là nhãn của thẻ hiện tại một số kí tự cuối cùng (ít nhất một kí tự). Số điểm nhận được thêm bằng số may mắn trên thẻ mới chọn;
2. Trong các thẻ ở phía bên phải thẻ hiện tại, chọn một thẻ mà số may mắn trên thẻ này và số may mắn trên thẻ hiện tại là một cặp số may mắn. Cụ thể, nếu số may mắn trên thẻ hiện tại là p_i và số may mắn trên thẻ mới chọn là p_j thì cặp số (p_i, p_j) hoặc (p_j, p_i) phải là một trong các cặp số may mắn mà Ban tổ chức đã công bố. Số điểm nhận được thêm bằng số may mắn trên thẻ mới chọn.

Sau một số bước chọn, nếu Nam không chọn được nữa thì số tiền thưởng bằng tổng điểm tích lũy được.

Yêu cầu: Hãy giúp Nam tìm cách chọn các thẻ để đạt được tổng số tiền thưởng là lớn nhất.

Dữ liệu

- Dòng thứ nhất chứa số nguyên dương n và số nguyên không âm m .
- Tiếp theo là n cặp dòng mô tả thông tin trên thẻ. Cặp dòng thứ i ($1 \leq i \leq n$) có dòng thứ nhất chứa nhãn và dòng thứ hai chứa số may mắn của thẻ thứ i .
- Mỗi dòng trong số m dòng cuối cùng chứa hai số nguyên dương mô tả một cặp số may mắn.

Các thẻ có thể có nhãn giống nhau và tổng số các kí tự của các nhãn trên các thẻ không vượt quá 10^6 . Các số may mắn có giá trị không vượt quá 10^5 . Các số trên cùng một dòng

cách nhau bởi dấu cách.

Kết quả

Ghi ra một số nguyên duy nhất là tổng tiền thưởng lớn nhất chọn được.

Ràng buộc

- Có 40% số test tương ứng với 40% số điểm của bài thỏa mãn: $n \leq 100$, nhãn trên các thẻ có độ dài không vượt quá 100 và $m \leq 100$;
- 40% số test khác ứng với 40% số điểm của bài thỏa mãn: $n \leq 3 \times 10^5$ và $m = 0$.
- 20% số test còn lại ứng với 20% số điểm của bài thỏa mãn: $n \leq 3 \times 10^5$ và $m \leq 3 \times 10^5$.

Ví dụ

Input	Output
4 1 ABA 5 BC 10 AB 5 A 6 6 10	16

Giải thích

Thứ tự thẻ	1	2	3	4
Nhãn thẻ	ABA	BC	AB	A
Số may mắn trên thẻ	5	10	5	6

Nam chọn thẻ thứ 2 rồi chọn tiếp thẻ thứ 4 bằng cách sử dụng cặp số may mắn (6, 10) để nhận được tổng tiền thưởng là 16.

Một cách chọn khác cũng được tổng tiền thưởng là 16 bằng cách chọn lần lượt các thẻ 1, 3, 4.

Problem AD. Nhân ma trận

Time Limit 2000 ms

Mem Limit 262144 kB

Ở một xã vùng cao có N hộ gia đình được đánh số từ 1 đến N . Do mạnh dạn áp dụng khoa học kỹ thuật vào làm kinh tế nên trong xã có nhiều hộ gia đình đã trở nên giàu có. Trong kỳ tổng kết cuối năm vừa qua, chính quyền xã biết được hộ gia đình thứ i ($1 \leq i \leq N$) có thu nhập là $A_i^{(0)}$. Hưởng ứng phong trào toàn dân làm kinh tế nâng cao thu nhập và thu hẹp khoảng cách giàu nghèo, bắt đầu từ năm nay, gọi là năm thứ nhất, hàng năm chính quyền xã yêu cầu với mỗi hộ gia đình, hộ thứ i ($1 \leq i \leq N$) phải nghiên cứu phương thức làm kinh tế năm trước của các hộ gia đình liên tiếp từ thứ L_i đến thứ R_i ($1 \leq L_i \leq R_i \leq N$) để học hỏi kinh nghiệm.

Với mỗi hộ gia đình X , nếu thu nhập năm trước lớn hơn hoặc bằng thu nhập của hộ có thu nhập cao nhất trong số các hộ gia đình mà hộ X phải học hỏi, thì hộ X giữ nguyên phương thức làm kinh tế cũ sao cho năm nay sẽ có thu nhập ổn định bằng năm trước. Trái lại, hộ X sẽ phải sửa đổi phương thức làm kinh tế với sự hỗ trợ của chính quyền sao cho thu nhập năm nay của hộ X sẽ phải bằng thu nhập cao nhất năm trước trong số các hộ mà hộ X học hỏi. Cụ thể, tại năm thứ t ($t \geq 1$), hộ gia đình thứ i ($1 \leq i \leq N$) sẽ có thu nhập là $A_i^{(t)} = \max(A_i^{(t-1)}, A_{L_i}^{(t-1)}, A_{L_{i+1}}^{(t-1)}, \dots, A_{R_i}^{(t-1)})$.

Yêu cầu: Hãy giúp chính quyền xác định bắt đầu từ năm thứ bao nhiêu thì mỗi hộ gia đình đều sẽ có thu nhập ổn định năm sau bằng năm trước.

Dữ liệu

Vào từ file văn bản **INCOME.INP**:

- Dòng đầu chứa một số nguyên dương N ($N \leq 3 \times 10^5$).
- Dòng thứ hai chứa N số nguyên dương $A_1^{(0)}, A_2^{(0)}, \dots, A_N^{(0)}$ ($A_i^{(0)} \leq 10^9, \forall i = 1, 2, \dots, N$).
- Dòng thứ i trong số N dòng tiếp theo chứa hai số nguyên dương L_i và R_i ($L_i \leq R_i \leq N$).

Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả

Ghi ra file văn bản **INCOME.OUT** một số nguyên t là năm mà bắt đầu từ đó trở đi, mỗi hộ gia đình đều có thu nhập ổn định năm sau bằng năm trước.

Ràng buộc

- Có 18% số test ứng với 18% số điểm thỏa mãn: $N \leq 500$.
- 16% số test khác ứng với 16% số điểm thỏa mãn: $A_i^{(50)} = A_i^{(49)}, \forall i = 1, 2, \dots, N$.
- 20% số test khác ứng với 20% số điểm thỏa mãn: $\sum_{i=1}^n (R_i - L_i) \leq 10^6$.
- 24% số test khác ứng với 24% số điểm thỏa mãn: Không tồn tại i, j ($1 \leq i, j \leq N$) thỏa mãn $L_i < L_j$ và $R_j < R_i$.
- 22% số test còn lại ứng với 22% số điểm không có ràng buộc gì thêm.

Ví dụ

Input	Output
4 1 2 3 4 2 2 3 3 1 2 1 3	3

Giải thích

- Ở năm đầu tiên, thu nhập các hộ lần lượt là (2, 3, 3, 4).
- Ở năm thứ hai, thu nhập các hộ lần lượt là (3, 3, 3, 4).
- Từ năm thứ ba trở đi, thu nhập của các hộ giữ nguyên là (3, 3, 3, 4).

Problem AE. Phở đèn đỏ

Time Limit 380 ms

Mem Limit 524288 kB

Đọc đề gốc [tại đây](#).

Một giàn đèn trang trí kích thước $m \times n$, các hàng đánh số từ 1 đến m từ trên xuống dưới, các cột đánh số từ 1 đến n từ trái sang phải. Ô nằm giao giữa hàng r và cột c gọi là ô (r, c) . Trên mỗi ô có một bóng đèn, mỗi bóng đèn có ba trạng thái: hoặc tắt, hoặc bật sáng màu xanh, hoặc bật sáng màu đỏ. Có k ô phân biệt $(r_1, c_1), (r_2, c_2), \dots, (r_k, c_k)$ của giàn đèn, mỗi ô có một công tắc điều khiển. Khi tác động vào công tắc của ô (r_t, c_t) thì những đèn nằm trong các ô thuộc hình chữ nhật có ô trái trên là (r_t, c_t) và ô phải dưới là (x_t, y_t) sẽ đổi trạng thái ($t = 1, 2, \dots, k$). Cụ thể, các đèn nằm trong các ô (u, v) mà $r_t \leq u \leq x_t$ và $c_t \leq v \leq y_t$ sẽ thay đổi theo quy tắc: nếu đèn đang ở trạng thái tắt sẽ chuyển sang trạng thái bật sáng màu xanh, nếu đang ở trạng thái bật sáng màu xanh sẽ chuyển sang trạng thái bật sáng màu đỏ, còn nếu đèn đang ở trạng thái bật sáng màu đỏ sẽ chuyển sang trạng thái tắt. Mỗi công tắc có thể tác động nhiều lần.

Yêu cầu: Cho thông tin trạng thái ban đầu các đèn trên giàn và các công tắc. Hãy tìm cách đưa tất cả các đèn về cùng một trạng thái bật sáng màu xanh hoặc bật sáng màu đỏ với số lần tác động là ít nhất.

Input

- Dòng đầu chứa ba số nguyên dương m, n, k ($k \leq m \times n$);
- Dòng thứ i trong số m dòng tiếp theo chứa n số nguyên nhận giá trị 0, 1 hoặc 2. Số thứ j ($j = 1, 2, \dots, n$) mô tả trạng thái đèn ở ô (i, j) là tắt, bật sáng màu xanh hoặc bật sáng màu đỏ tương ứng với các giá trị 0, 1 hoặc 2;
- Dòng thứ t trong số k dòng tiếp theo chứa bốn số nguyên dương r_t, c_t, x_t, y_t ($1 \leq r_t \leq x_t \leq m$ và $1 \leq c_t \leq y_t \leq n$).

Các số trên cùng một dòng cách nhau bởi dấu cách.

Output

Ghi ra duy nhất một số nguyên là số lần tác động ít nhất để các đèn về cùng một trạng thái sáng màu xanh hoặc sáng màu đỏ, nếu không tồn tại cách tác động thì ghi số -1 .

Giới hạn

- Có 40% số test ứng với 40% số điểm của bài thỏa mãn điều kiện: $m = 1; n \leq 10$;
- 40% số test khác ứng với 40% số điểm của bài thỏa mãn điều kiện: $m \times n \leq 10^3$;
- 20% số test còn lại ứng với 20% số điểm của bài thỏa mãn điều kiện: $m \times n \leq 10^5$.

Input	Output
2 3 3 2 1 0 2 1 0 1 1 2 3 1 2 2 3 1 3 2 3	2

Note

Trước tiên tác động vào công tắc ở ô (1, 3), sau đó tác động vào công tắc ở ô (1, 2), khi đó tất cả các đèn trên giàn đều sáng màu đỏ.

Problem AF. Quy hoạch động dãy con chung

Time Limit 1000 ms

Mem Limit 524288 kB

Dãy $C = c_1, c_2, \dots, c_k$ được gọi là dãy con của dãy $A = a_1, a_2, \dots, a_n$ nếu C có thể nhận được bằng cách xóa bớt một số phần tử của dãy A và giữ nguyên thứ tự của các phần tử còn lại, nghĩa là tìm được dãy các chỉ số $1 \leq l_1 < l_2 < \dots < l_k \leq n$ sao cho $c_1 = a_{l_1}, c_2 = a_{l_2}, \dots, c_k = a_{l_k}$. Ta gọi độ dài của dãy là số phần tử của dãy.

Cho hai dãy $A = a_1, a_2, \dots, a_m$ và $B = b_1, b_2, \dots, b_n$. Dãy $C = c_1, c_2, \dots, c_k$ được gọi là dãy con chung bội hai của dãy A và B nếu C vừa là dãy con của dãy A , vừa là dãy con của dãy B và thỏa mãn điều kiện $2 \times c_i \leq c_{i+1}$ ($i = 1, 2, \dots, k - 1$).

Cho hai dãy A và B . Hãy tìm độ dài dãy con chung bội hai có độ dài lớn nhất của hai dãy A và B .

Input

Dòng đầu tiên chứa T là số lượng bộ dữ liệu. Tiếp đến là T nhóm dòng, mỗi nhóm cho thông tin về một bộ dữ liệu theo khuôn dạng sau:

- Dòng đầu chứa 2 số nguyên dương m và n .
- Dòng thứ hai chứa m số nguyên không âm a_1, a_2, \dots, a_m mỗi số không vượt quá 10^9 .
- Dòng thứ ba chứa n số nguyên không âm b_1, b_2, \dots, b_n mỗi số không vượt quá 10^9 .
- Các số trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.

Output

Ghi ra T dòng, mỗi dòng ghi một số nguyên là độ dài dãy con chung bội hai dài nhất của dãy A và B tương ứng với bộ dữ liệu vào.

Giới hạn

- 30% số test có $m, n \leq 15$.
- 30% số test khác có $m, n \leq 150$.
- Có 40% số test còn lại có $m, n \leq 1500$.

Input	Output
1 5 5 5 1 6 10 20 1 8 6 10 20	3

Problem AG. Dynamic Connectivity

Time Limit 1200 ms

Mem Limit 524288 kB

Đọc đề gốc [tại đây](#).

Xe buýt là một phương tiện giao thông phổ biến tại thành phố mà Alice sinh sống bởi tính tiện dụng và giá cả hợp lý của nó. Thành phố có n bến xe buýt được đánh số từ 1 tới n và có m tuyến xe buýt hai chiều, mỗi tuyến đang được điều hành bởi một trong hai công ty vận tải A hoặc B . Cụ thể, tuyến thứ i ($1 \leq i \leq m$) di chuyển giữa hai bến u_i và v_i với giá vé do công ty quản lý quy định là w_i .

Lưu ý là giữa hai bến có thể có nhiều hơn một tuyến xe buýt.

Hai công ty A và B đều có chính sách giảm giá vé cho những ai thường xuyên đi bằng xe buýt. Cụ thể, mỗi ngày công ty A sẽ chỉ thu số tiền bằng với giá vé lớn nhất trong tất cả các tuyến được điều hành bởi công ty A mà khách hàng đã đi trong ngày. Để cạnh tranh, công ty B cũng có chính sách tương tự: khách hàng sẽ chỉ phải trả số tiền bằng giá vé lớn nhất trong tất cả các tuyến thuộc công ty B mà người đó đã đi trong ngày.

Nhà Alice ở gần bến xe buýt s và nơi làm của Alice ở gần bến xe buýt t nên hàng ngày Alice đều phải đi lại giữa hai bến này thông qua các tuyến xe buýt.

Yêu cầu: Bạn hãy giúp Alice xác định số tiền nhỏ nhất cần bỏ ra mỗi ngày để đảm bảo việc đi từ bến xe buýt s đến bến xe buýt t .

Input

- Dòng đầu tiên chứa bốn số nguyên dương n, m, s và t ($n, m \leq 50\,000$; $s, t \leq n$; $s \neq t$);
- Dòng thứ i ($1 \leq i \leq m$) trong m dòng tiếp theo chứa bốn số nguyên dương c_i, u_i, v_i, w_i ($u_i, v_i \leq n$; $u_i \neq v_i$; $w_i \leq 10^9$) mô tả tuyến xe buýt thứ i trong đó $c_i = 1$ nếu tuyến này được điều hành bởi công ty A hoặc $c_i = 2$ nếu tuyến này được điều hành bởi công ty B .

Các số trên cùng một dòng cách nhau bởi dấu cách. Dữ liệu đảm bảo luôn tồn tại cách đi lại giữa hai bến xe buýt s và t thông qua m tuyến xe.

Output

Ghi ra một số nguyên duy nhất là số tiền nhỏ nhất cần bỏ ra mỗi ngày để đảm bảo được việc đi từ bến xe buýt s đến bến xe buýt t cho Alice.

Giới hạn

- Có 20% số lượng test ứng với 20% số điểm của bài thỏa mãn điều kiện: tất cả m tuyến xe buýt đều được điều hành bởi công ty A ;
- 30% số lượng test khác ứng với 30% số điểm của bài thỏa mãn điều kiện: $n, m \leq 5\,000$;
- 20% số lượng test khác ứng với 20% số điểm của bài thỏa mãn điều kiện: tồn tại cách đi xe buýt tối ưu trong đó Alice chỉ sử dụng tối đa một tuyến xe của công ty B ;
- 30% số lượng test còn lại ứng với 30% số điểm của bài không có điều kiện gì thêm.

Input	Output
6 7 1 4 1 1 2 4 2 2 3 7 1 3 4 6 2 1 6 5 1 6 5 5 2 5 4 8 2 2 5 2	12

Note

Để đi từ 1 đến 4, Alice sẽ lần lượt đi tuyến (1, 2) của công ty A và hai tuyến (2, 5), (5, 4) của công ty B . Khi đó số tiền mà Alice phải trả cho công ty A là 4 và trả cho công ty B là 8.

