Problem Wood. Alice, Bob, and two arrays.

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2.5 seconds
Memory limit:	1024 megabytes

There is an array a of length N and an array b of length M. All numbers in the arrays are integers and lie in the range from 1 to k. There is also an initially empty array c.

Alice and Bob play the following game on these arrays: the players take turns, and on their turn, a player must append a number to the end of the array c such that c remains a subsequence of both a and b. The player who cannot make a move loses. Alice goes first.

They will play the game q times. For the i-th game, they will choose two numbers x_i and y_i ($0 \le x_i < N, 0 \le y_i < M$), then remove the first x_i elements from array a and the first y_i elements from array b, and then play the game on the resulting arrays. After each deletion operation and before the next one, the arrays a and b are restored to their initial state, meaning that the numbers deleted from the arrays in one game may not be deleted in subsequent games. Also, the array c is cleared between games.

The guys have their quirks, so they always choose x_i and y_i in such a way that after the deletions, **the remaining parts of arrays a and b start with the same value**.

Alice really wants to win, so she asks you to determine for each game whether she can win, assuming both players play optimally.

Note that the arrays can be very long, so they are provided in a special format. Each array is given as a sequence of segments of equal numbers. Array a consists of n such segments, and array b consists of m such segments. Each segment is defined by its length and the number that occupies that segment.

Input

The first line contains six integers N, n, M, m, k, q ($1 \le N, M \le 10^9$, $1 \le n, m, k \le 1600$, $1 \le q \le 10^6$) — the length of the first array, the number of segments in the first array, the length of the second array, the number of segments in the second array, the number limit, and the number of games, respectively.

The next n lines contain two integers l_i^a and v_i^a ($1 \le l_i^a \le N$, $1 \le v_i^a \le k$) — the length of the segment and the number written in that segment. These numbers define array a: the first l_1^a numbers of array a are equal to v_1^a, the next l_2^a numbers are equal to v_2^a, …, the last l_n^a numbers are equal to v_n^a.

The next m lines contain two integers l_i^b and v_i^b ($1 \le l_i^b \le N$, $1 \le v_i^b \le k$) — the length of the segment and the number written in that segment. These numbers define array b. The format is similar to array a.

It is guaranteed that $\sum l_i^a = N$, $\sum l_i^b = M$, $v_i^a \ne v_{i+1}^a$, and $v_i^b \ne v_{i+1}^b$.

The next q lines contain pairs of integers x_i and y_i ($0 \le x_i < N$, $0 \le y_i < M$) — descriptions of the games.

For each game i, it is guaranteed that if we remove the first x_i elements from a and the first y_i elements from b, the remaining parts of the arrays will start with the same value.

Output

For each of the q games, print "Yes" if Alice wins with optimal strategy, and "No" if Bob wins.

Examples

input	output
5 1 5 1 1 9 5 1 5 1 0 0 0 1 0 2 1 0 1 1 1 2 2 0 2 1 2 2	Yes No Yes No No Yes Yes Yes Yes
7 3 7 3 2 12 2 1 3 2 2 1 2 2 3 1 2 2 0 2 0 3 0 4 1 2 1 3 1 4 2 5 2 6 3 5 3 6 4 5 4 6	Yes No Yes Yes No Yes No Yes No Yes Yes Yes

Note

In the first example, the arrays look like this: $a = (1, 1, 1, 1, 1)$ and $b = (1, 1, 1, 1, 1)$.

- In the first query, $x = 0, y = 0$, so the game will be played on the arrays $a = (1, 1, 1, 1, 1)$ and $b = (1, 1, 1, 1, 1)$. In this case, players can only append the number 1 to the array c, so after 5 moves the game will end, and Bob will lose because he won't be able to make a move.
- In the second query, $x = 0, y = 1$, so the game will be played on the arrays $a = (1, 1, 1, 1, 1)$ and $b = (1, 1, 1, 1)$. In this case, the game will end after 4 moves, and Alice will lose.
- In the last query, $x = 2, y = 2$, so the game will be played on the arrays $a = (1, 1, 1)$ and $b = (1, 1, 1)$. In this case, Bob will lose.

In the second example, $a = (1, 1, 2, 2, 2, 1, 1)$, $b = (2, 2, 1, 1, 1, 2, 2)$.

- In the first query, $x = 0$ and $y = 2$, so the game will be played on the arrays $a = (1, 1, 2, 2, 2, 1, 1)$ and $b = (1, 1, 1, 2, 2)$. If Alice appends the number 2 to the array c, Bob will also append the number 2, and then no moves will be left, so Alice will lose. Therefore, Alice must first append the number

1 to c. After this, for similar reasons, if Bob appends the number 2 to the array, he will lose. So, he is forced to append the number 1, and the array c becomes $(1, 1)$. Then Alice again appends the number 1 to the array c, and Bob has no more moves left, so Alice wins.

- In the second query, $x = 0$ and $y = 3$, so the game will be played on the arrays $a = (1, 1, 2, 2, 2, 1, 1)$ and $b = (1, 1, 2, 2)$. Following the reasoning in the previous example, Alice cannot append the number 2 to the array c, because then she will lose. But if Alice appends the number 1, then Bob will also append the number 1, and after that, Alice will lose for similar reasons. Therefore, Bob wins in this case.

Scoring

The tests for this problem consist of eleven groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

Group	Points	Additional constraints			Required Groups	Comment
		N, M	n, m	q		
0	0	–	–	–	–	Examples.
1	13	$N, M \le 300$	–	$q \le 10^5$	0	
2	12	$N, M \le 5000$	–	$q \le 10^5$	0, 1	
3	11	–	–	$q \le 10^5$	–	$l_i^a \le 1000$ and all v_i^a are distinct $l_i^b \le 1000$ and all v_i^b are distinct
4	8	–	–	$q \le 10^5$	3	$l_i^a \le 1000$ and all v_i^a are distinct
5	10	–	–	$q \le 10^5$	–	$l_1^a \ge N - 500$ and $v_1^a = 1$ $l_1^b \ge M - 500$ and $v_1^b = 1$
6	7	$N, M \le 10^5$	$n, m \le 100$	$q \le 10^5$	–	$k \le 5$
7	6	$N, M \le 10^5$	$n, m \le 100$	$q \le 10^5$	0, 6	$k \le 50$
8	7	–	$n, m \le 100$	$q \le 10^5$	0, 6, 7	$k \le 50$
9	9	–	$n, m \le 800$	$q \le 10^5$	0, 6 – 8	
10	10	–	–	$q \le 10^5$	0 – 9	**Offline-evaluation.**
11	7	–	–	–	0 – 10	**Offline-evaluation.**

Problem Stone. The arithmetic exercise

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Oleg and Dasha participated in a team competition, but unfortunately, they were unable to solve any problems. Oleg immediately realized that their team wasn't training enough. Then, their mutual friend suggested an interesting exercise. The exercise was quite simple, and to solve it, they only needed to know the rules of addition and subtraction of integers.

You are given an array a of length n, where initially all values are zero. You are also given m numbers $x_1, x_2, \ldots x_m$. Then, for each i from 1 to m, you choose some index j_i and make the change $a_{j_i} = x_i - a_{j_i}$.

Help Oleg and Dasha determine what the maximum possible sum of the elements of array a can be after all the changes, if the choices are made optimally.

Input

Each test consists of several input data sets. The first line contains a single integer t $(1 \le t \le 10\,000)$ — the number of input data sets. Then follows the description of the data sets.

The first line of each data set contains two integers n and m $(1 \le n, m \le 300\,000)$ — the length of the array a and the number of values x_i, respectively.

The second line of each data set contains m integers $x_1, x_2, \ldots, x_m$ $(-10^9 \le x_i \le 10^9)$ — the description of the values.

Let N be the sum of n over all data sets, and M be the sum of m over all data sets.

It is guaranteed that N and M do not exceed 300 000.

Output

For each data set, output a single number on a new line — the maximum sum of the array a that can be obtained.

Example

input	output
4 1 4 1 2 3 4 2 7 10 3 7 1 4 6 3 4 10 103 354 1 227 179 189 142 201 165 140 5 3 -10 11 -4	2 18 1085 17

Note

In the first data set, all operations are applied to the first element of the array a. It sequentially becomes $1 - 0 = 1$, $2 - 1 = 1$, $3 - 1 = 2$, $4 - 2 = 2$, so the answer is 2.

In the second data set, the following sequence of changes can be performed:

1. Apply the change to the first element: $a_1 = 10 - a_1 = 10 - 0 = 10$, $a = [10, 0]$.
2. Apply the change to the first element: $a_1 = 3 - a_1 = 3 - 10 = -7$, $a = [-7, 0]$.

3. Apply the change to the first element: $a_1 = 7 - a_1 = 7 - (-7) = 14$, $a = [14, 0]$.
4. Apply the change to the first element: $a_1 = 1 - a_1 = 1 - 14 = -13$, $a = [-13, 0]$.
5. Apply the change to the second element: $a_2 = 4 - a_2 = 4 - 0 = 4$, $a = [-13, 4]$.
6. Apply the change to the first element: $a_1 = 6 - a_1 = 6 - (-13) = 19$, $a = [19, 4]$.
7. Apply the change to the second element: $a_2 = 3 - a_2 = 3 - 4 = -1$, $a = [19, -1]$.

At the end, we have $a = [19, -1]$, so the final sum is 18.

It can be shown that a better result is not possible.

Scoring

The tests for this problem consist of ten groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

Group	Points	Additional constraints			Required Groups	Comment
		n, N	m, M	x_i		
0	0	–	–	–	–	Examples.
1	4	–	–	$0 \le x_i$	–	All x_i are same
2	8	$n = 2$	$M \le 30$ $m \le 18$	–	–	
3	11	$n = 2$	$M \le 50$	$-10 \le x_i \le 10$	–	
4	9	$n = 2$	$M \le 400$	$-400 \le x_i \le 400$	3	
5	8	$N \le 30$ $n \le 18$	$M \le 30$ $m \le 18$	–	0	
6	10	$N \le 2000$	$M \le 2000$	$0 \le x_i$	–	
7	12	$N \le 2000$	$M \le 2000$	–	0, 2 – 6	
8	10	–	–	$0 \le x_i$	1	There are no more than two different values among x_i
9	17	–	–	$0 \le x_i$	1, 6, 8	
10	11	–	–	–	0 – 9	**Offline-evaluation.**

Problem Iron. Dreaming is not harmful

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

In the wedding agency M., mass layoffs are planned. All employees are busy counting the days until, with a favorable turn of events, they can head the company instead of working.

The structure of the company represents a suspended tree with the root at vertex number 1. The immediate supervisor of employee number v is the employee with number p_v. The level of competence of employee v is defined by the parameter s_v. This parameter is different for all employees. The higher the level of competence, the more useful the employee is to the company. Note that as a result of an opaque hiring process, it may happen that a less competent employee is the supervisor of a more competent one.

As a result of significant personnel restructuring, every day the CEO, who is at the root of the working hierarchy, will be fired. If there are employees left in the company, the most competent immediate subordinate will take their place. After that, the other subordinates of the former director will become subordinates of the new director. See the explanations in the examples for a better understanding of the condition.

Each employee easily calculated how many days it would take for them to become the CEO. Many were not ready to wait that long, as they would only get to be the director for one day! To speed up this process, they are ready to "cancel" one of their colleagues. The "canceled" employee's level of competence drops to 0, as no one is willing to interact with them anymore.

You will need to answer q queries. In the k-th query, employee number v_k is interested in the minimum number of days until they can head the company if they are willing to "cancel" exactly one employee. All queries are imaginary and independent, and the real levels of competence of the employees remain unchanged for all queries.

Input

The first line contains two integers n, q ($2 \le n \le 300\,000$, $1 \le q \le n$) — the number of employees and the number of queries.

The second line contains $n-1$ integers $p_2, p_3, \ldots, p_n$ ($1 \le p_i < i$) — the immediate supervisors of employees numbered from 2 to n.

The third line contains n integers $s_1, s_2, \ldots, s_n$ ($1 \le s_i \le n$) — the levels of competence of the employees. It is guaranteed that they are all different.

The fourth line contains q integers $v_1, v_2, \ldots, v_q$ ($1 \le v_i \le n$) — the promotion queries. It is guaranteed that all numbers v_i are distinct.

Output

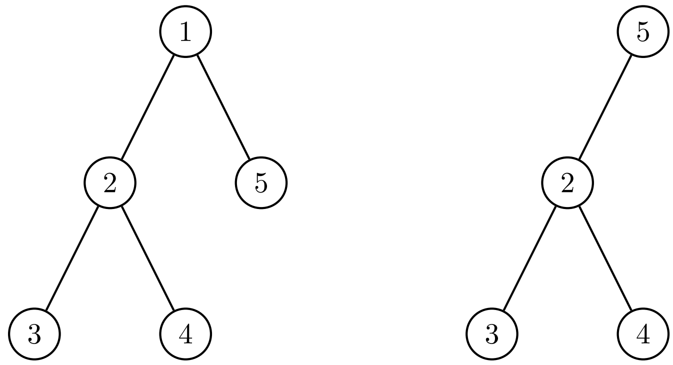
Output q integers separated by spaces — the minimum number of days after which employees $v_1, v_2, \ldots, v_q$ can become directors.

Example

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

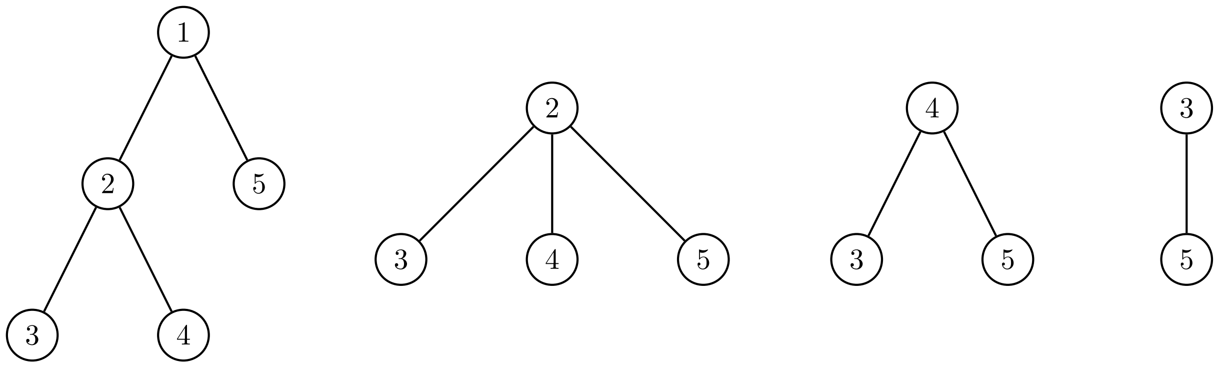
Note

In the test example, the fifth employee can head the company in 1 day. To do this, it is enough to "cancel" the second employee. The structure of the company will change as follows:



Initial structure   Day 1

The third employee can head the company in 3 days. To do this, it is enough to "cancel" the fifth or fourth employee. If the fifth is canceled, the structure of the company will change as follows:



Initial structure   Day 1   Day 2   Day 3

The first employee is already the head of the company, so the answer to the corresponding query is 0.

The fourth employee can become the head of the company in two days. It is enough, similarly to the previous example, to "cancel" the fifth employee.

Scoring

The tests for this problem consist of nine groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

The table with the groups is on the next page.

Group	Points	Additional constraints		Required Groups	Comment
		n	q		
0	0	–	–	–	Examples.
1	10	–	–	–	$p_i = 1$ or $p_i = i - 1$, with $p_i = 1$ for no more than two numbers i
2	6	–	–	1	$p_i = 1$ or $p_i = i - 1$
3	8	$n \le 50$	$q \le 50$	0	
4	13	$n \le 1000$	$q \le 1000$	0, 3	
5	11	–	$q \le 100$	0, 3	
6	9	–	–	–	$p_i = \lfloor \frac{i}{2} \rfloor$
7	11	–	–	0, 3, 6	The number of supervisors* for any employee does not exceed 100
8	14	–	–	–	$s_i > s_{p_i}$ for any $i > 1$
9	18	–	–	0 – 8	**Offline-evaluation.**

Supervisors* of an employee — the set of their immediate supervisor and all supervisors of their immediate supervisor.

Problem Diamond. Cute Subsequences

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	256 megabytes

You are given an array of n positive integers $a_1, a_2, \ldots, a_n$, as well as a positive integer k. You need to divide the array into k non-empty subsequences such that each element of the array belongs to exactly one subsequence. A subsequence is a sequence that can be obtained from another sequence by deleting some elements without changing the order of the remaining elements.

Let the i-th subsequence contain elements with indices $j_1 < \ldots < j_l$. The *value* of this subsequence is defined as the maximum value of $a_{j_m} + m$ for all m from 1 to l.

The *cost* of dividing the array into k subsequences is the sum of the *values* of these subsequences.

Find the maximum *cost* of the division.

Input

The first line contains two positive integers n and k ($1 \le k \le n \le 500\,000$) — the size of the array and the number of subsequences to divide it into.

The second line contains n positive integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the array.

Output

Output the maximum *cost* of dividing the given array into k non-empty subsequences.

Example

input	output
5 3 3 7 10 1 2	24

Note

In the sample test, the array can be divided into $[3, 10]$, $[7]$, $[1, 2]$. Then the answer will be $(10 + 2) + (7 + 1) + (2 + 2) = 12 + 8 + 4 = 24$.

Scoring

The tests for this problem consist of six groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed.

Group	Points	Additional Constraints		Required Groups	Comment
		n	k		
0	0	–	–	–	Examples.
1	14	$n \le 8$	–	0	
2	19	–	$k = 2$	–	
3	17	–	–	–	$a_{i+1} \le a_i$
4	21	–	–	–	$a_{i+1} \ge a_i - 1$
5	15	$n \le 1000$	–	0, 1	
6	14	–	–	0 – 5	