John's Gardening Invention (gardening)

John the programmer is burnt out after working in the IT sector for so long, so he started to learn about gardening. The current topic he's focusing on is titled "The watering of plants".

Figure 1: No more manual labor!

John created a watering system which can be modeled as a rooted tree graph, consisting of N junctions (numbered from 0 to $N-1$) and $N-1$ pipes. Each pipe connects two different junctions, and it is possible to get from any junction to any other junction through the pipes. There's a *main junction* where the water is pumped into the system.

The layout of the system can be described by N integers: $P_0, P_1, \ldots, P_{N-1}$, where $P_i = -1$ if junction i is the main junction, otherwise there's a pipe connecting junctions i and P_i.

Every junction has one of two possible states: *open* or *closed*, which can be toggled by rotating a valve at the junction. Initially, junction i is in state S_i. The cost of toggling the state of junction i is C_i. The water will reach a junction if *every* junction on the path from the main junction to it is open.

John has put plants at every junction x $(0 \le x \le N-1)$ for which there's no junction j having $P_j = x$. The water requirements of plants are different: some **must** be watered, some **must not** be.

John has Q queries: in the i-th one he gives you the indices $A_1, A_2, \ldots, A_{M_i}$ of M_i junctions with plants. You should answer these queries by calculating the **minimum** cost to change the states of the junctions, so that all these plants are watered, but no other plant is. All queries are independent, i.e., the state of the junctions don't change after a query.

> ☞ Among the attachments of this task you may find a template file `gardening.*` with a sample incomplete implementation.

Input

The first line contains two integer N and Q.

The second line contains N integers: $P_0, P_1, \ldots, P_{N-1}$, the description of the watering system.
The third line contains N integers: $C_0, C_1, \ldots, C_{N-1}$, the cost to change the state of each junction.

The fourth line contains N integers: $S_0, S_1, \ldots, S_{N-1}$, the initial state of each junction (1 if open, 0 if closed).

The next $2Q$ lines contain the description of the queries. The $(2i-1)$-th line contains M_i and the $(2i)$-th contains: $A_{i,1}, A_{i,2}, \ldots, A_{i,M_i}$, the indices of junctions with plants that should be watered.

Output

You need to write Q integers, each on a separate line: the answers to queries.

Constraints

- $2 \leq N \leq 200\,000$.
- $1 \leq Q \leq 200\,000$.
- $-1 \leq P_i \leq N-1$ for all $0 \leq i \leq N-1$ and there's only one node with $P_i = -1$.
- $1 \leq C_i \leq 10^9$ for all $0 \leq i \leq N-1$.
- $S_i = 0$ or $S_i = 1$ for all $0 \leq i \leq N-1$.
- $M_i < N$ and $\sum M_i \leq 500\,000$.
- $0 \leq A_{i,j} \leq N-1$ for all $1 \leq j \leq M_i$.
- $A_{i,j}$ is a junction with a plant.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (8 points) $N \leq 3$ and $\sum M_i \leq 10$.

– **Subtask 3** (11 points) $N \leq 16$ and $\sum M_i \leq 256$.

– **Subtask 4** (30 points) $N \leq 5000$ and $\sum M_i \leq 10\,000$.

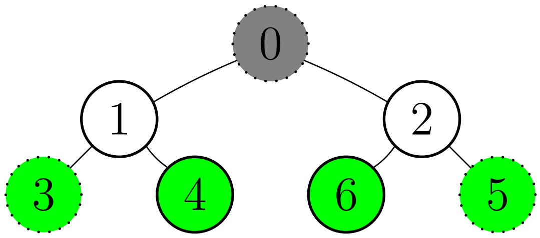
– **Subtask 5** (51 points) No additional limitations.

Examples

input	output
7 3 -1 0 0 1 1 2 2 1 2 3 4 5 6 7 0 1 1 0 1 0 1 4 3 4 5 6 1 3 2 3 5	11 13 23

Explanation

The initial state of the system is displayed below. Green nodes are junctions with plants, and the grey node is the main junction. Dotted nodes represent closed junctions, the others junctions are open.



In the **first query**, junctions $3, 4, 5$ and 6 must be watered. For this, all (initially closed) junctions should be opened for a cost of $1 + 4 + 6 = 11$.

In the **second query**, only junction 3 must be watered. Junctions 0 and 3 should be opened, while 2 and 4 should be closed to prevent the watering of any other plant. The total cost is $1 + 4 + 3 + 5 = 13$. Note that closing junction 6 instead of junction 4 achieves the same goal, but at a higher cost.