Traveling in Cells

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

There are n cells arranged in a row. The i-th cell has a color c_i and contains a ball with value v_i.

You're going to travel several times in the cells. For each travel, you'll be given an integer x and a set of colors $\mathbb{A} = \{a_1, a_2, \cdots, a_k\}$ where $c_x \in \mathbb{A}$. The travel starts from cell x. During the travel, if you're located in cell i you can next move to cell $(i-1)$ or $(i+1)$. Note that you can't move out of these n cells. Also at any time, the color of cell you're located in must belong to set $\mathbb{A}$.

When you're in cell i, you can choose to remove the ball in the cell and gain its value v_i. As there is only one ball in each cell, you can only remove the ball from each cell once.

Your task is to process q operations in order. Each operation is one of the following three types:

- $1\ p\ x$: Change c_p to x.
- $2\ p\ x$: Change v_p to x.
- $3\ x\ k\ a_1\ a_2\ \ldots\ a_k$: Given the starting cell x and the color set $\mathbb{A} = \{a_1, a_2, \cdots, a_k\}$ of a travel, imagine that you're going on this travel, calculate the maximum total value you can gain. Note that this travel is only an imagination, thus the balls won't be truely removed. That is, all queries are independent.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and q ($1 \le n \le 10^5$, $1 \le q \le 10^5$) indicating the number of cells and the number of operations.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le n$) where c_i is the initial color of the i-th cell.

The third line contains n integers $v_1, v_2, \ldots, v_n$ ($1 \le v_i \le 10^9$) where v_i is the initial value of ball in the i-th cell.

For the following q lines, the i-th line describes the i-th operation. The input format is listed as follows:

- $1\ p\ x$: $1 \le p \le n$ and $1 \le x \le n$.
- $2\ p\ x$: $1 \le p \le n$ and $1 \le x \le 10^9$.
- $3\ x\ k\ a_1\ a_2\ \ldots\ a_k$: $1 \le x \le n$, $1 \le a_1 < a_2 < \ldots < a_k \le n$ and $c_x \in \{a_1, a_2, \cdots, a_k\}$.

It's guaranteed that neither the sum of n nor the sum of q of all test cases will exceed 3×10^5. Also the sum of k of all test cases will not exceed 10^6.

Output

For each operation of type 3 output one line containing one integer, indicating the maximum total value you can gain.

Example

standard input	standard output
2 5 10 1 2 3 1 2 1 10 100 1000 10000 3 3 1 3 3 3 2 2 3 2 5 20000 2 3 200 3 3 2 1 3 3 3 3 1 2 3 1 3 4 2 1 100000 1 2 2 3 1 2 1 2 4 1 1 2 3 4 1000000 1000000 1000000 1000000 3 4 4 1 2 3 4	100 110 1200 21211 100010 4000000