Problem I. Equivalence in Connectivity

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Two undirected graphs of size n are equivalent in connectivity when there is a path from u to v in one graph if and only if there is a path from u to v in the other graph for all $1 \le u < v \le n$.

Given is a sequence of k graphs $G_1, G_2, \ldots, G_k$. Each graph is of size n. In this sequence, for each $i = 2, 3, \ldots, k$, there exists $p_i < i$ such that G_i can be obtained from G_{p_i} by adding or removing an edge. Divide the given graphs into groups: two graphs must be in the same group if and only if they are equivalent in connectivity.

Input

There are multiple test cases. The first line of input contains an integer T ($1 \le T \le 10^5$), the number of test cases. For each test case:

The first line contains three integers k, n, and m ($1 \le k, n \le 10^5$, $0 \le m \le \min\left(10^5, \frac{n(n-1)}{2}\right)$): the number of graphs, the number of vertices in each graph, and the number of edges in G_1.

Each of the following m lines contains two integers u and v ($1 \le u < v \le n$), denoting an edge of G_1 connecting u and v. It is guaranteed that there are no multiple edges in G_1.

The i-th of the following $k-1$ lines contains an integer p_{i+1}, a string t_{i+1}, and two integers x_{i+1} and y_{i+1} ($1 \le p_{i+1} \le i$, $1 \le x_{i+1} < y_{i+1} \le n$). Each string t_{i+1} is either "`add`" or "`remove`".

If t_{i+1} is "`add`", then G_{i+1} is obtained from $G_{p_{i+1}}$ by adding an edge connecting x_{i+1} and y_{i+1}. It is guaranteed that this edge does not exist in $G_{p_{i+1}}$.

If t_{i+1} is "`remove`", then G_{i+1} is obtained from $G_{p_{i+1}}$ by removing an edge connecting x_{i+1} and y_{i+1}. It is guaranteed that this edge exists in $G_{p_{i+1}}$.

It is guaranteed that the sum of n, the sum of m, and the sum of k in all test cases do not exceed 10^5.

Output

For each test case:

On the first line, output an integer r: the number of groups.

For each group, output a single line which contains an integer k followed by k integers: the size of the group and the numbers of graphs in the group.

You can output the groups and the graphs in any order.

Example

standard input	standard output
2 15 11 8 6 11 1 6 6 9 6 8 1 2 1 5 9 10 2 5 1 add 3 11 1 add 2 3 3 add 5 8 4 add 5 11 3 add 7 10 1 add 6 10 3 add 3 10 1 remove 6 8 5 add 4 9 1 add 2 9 8 add 7 8 3 add 2 4 1 remove 6 9 10 remove 6 9 14 5 2 1 5 1 4 1 add 2 4 1 add 3 4 1 add 2 4 4 add 3 4 4 add 1 3 5 add 1 3 2 add 2 3 1 add 1 2 4 add 3 4 3 add 4 5 9 add 2 3 3 remove 1 5 3 remove 3 4	7 2 10 13 5 2 3 4 5 8 3 1 7 11 1 14 2 6 12 1 9 1 15 5 3 2 4 9 6 5 6 7 8 10 12 2 1 14 2 3 11 1 13