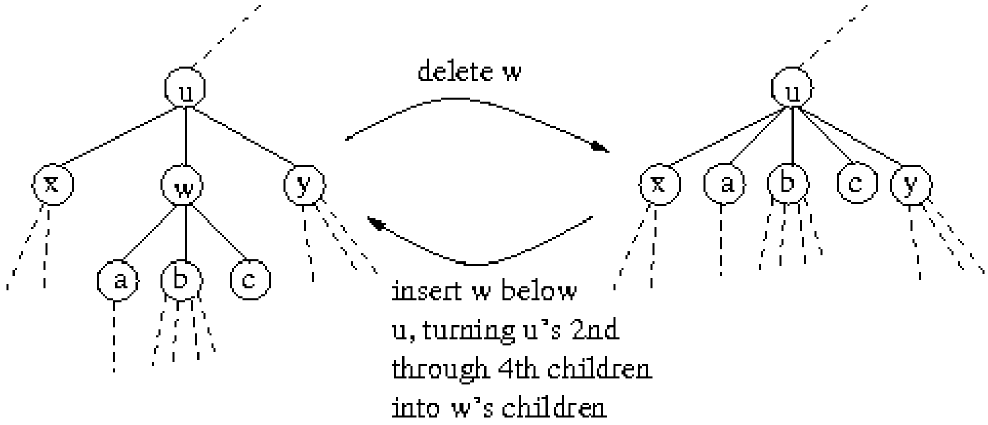
Tree Similarity

You are given two labeled and ordered rooted trees T and T' and would like to calculate the *distance* from T to T', which is the minimum number of operations you can perform on T to make it *equivalent* to T'. For each operation you can choose to do one of three things:

1. change the label of one node in T
2. delete a non-root node in T
3. insert a new node in T at a position somewhere below its root

Recall the trees T and T' are ordered, which means that if a non-leaf node has c children, its children are ordered from 1 to c. That is, there is a 1st child, a 2nd child, etc., all the way up to a cth child. When we say a tree X is equivalent to a tree Y, we mean the root of X should have the same label as the root of Y, their roots should have the same number of children (call it c), and the subtree rooted at the ith child of the root of X should be equivalent to the subtree rooted at the ith child of the root of Y for i=1,2,...,c. We now describe what we mean by deletion and insertion of non-root nodes in T. When deleting a non-root node w with d children, let u be its parent and suppose w is u's ith child. Then the first child of w becomes u's ith child, the second child of w becomes u's (i+1)st child, etc. For j < i, the jth child of u remains the same, but for all j > i, the child which was formerly the jth child of u now becomes its (j+d-1)st child (they get "shifted over" due to the insertion of w's children into u's child list). To insert a non-root node w into the tree, we can choose any node u to be its parent, and we can choose any contiguous subsequence (possibly empty) of u's children to become w's children, putting w in their place. When inserting a node, we can give it any label we want at the time of insertion. The root of T can never be deleted, and you can never insert a new node above the root to become the old root's parent. You can, however, change the label of the root.



Input

The first line contains n and m separated by a space, the sizes of the trees T and T', respectively (1 ≤ n,m ≤ 60). The next n lines describe T. On the ith line is a description of the ith node in the tree: its label, the number of children it has, then a list of its children in order from first to last, all space-separated. The next m lines similarly describe T'. Labels are nonnegative integers fitting in a 32-bit signed integer. The root of each tree is the node which is not the child of any other node in the tree.

Output

On a single line output the minimum number of operations that can be performed on T to make it equivalent to T', followed by a newline.

Example

Input:

```
3 2
6 0
1 2 0 2
4 0
2 1 1
4 0
```

Output:

```
2
```