Easy Password

This is the year 2100. Pranjali ma'am due to her intelligence survived this far, by her "Secret Potion" to freeze her age. She keeps her lotion in a very secret vault which is electronic password protected. This is a very special system designed by our great Sameer sir, which don't take any specific string as a password. Rather than it takes a "Tree", yes you heard that right a "Tree" of numbers as a password. Your password should be exact to the tree both in structure and value to unlock the vault. Its high time and many hackers are trying to hack the system. So Pranjali ma'am was bit worried and wanted to change the password. Some of the new passwords are very vulnerable so Sameer sir told not to use them. You have to make the **new password tree** for Pranjali ma'am to keep her "Secret Potion" safe. You are given **old password tree**, and the one **restricted password tree** provided by Sameer sir. The structure and value of both might not be same as Sameer sir don't know the exact password, he only know the current hacking pattern of the hackers.

You have to make a **new password tree** with same structure as **old password tree**, with the nodes of **old password tree** permuted in some order and any new node should not be exactly equal in position and value to **restricted password tree**. The position of a node is defined by the sequence of moves (Left / Right) required to go down from the root to that node.

If there are multiple answers, print the one whose difference is minimum from the **old password tree**.

Calculating difference between trees :

Diff = |sum((old[node[i]] - new[node[i]])*10^i)| where i is n-1 to 0. The ordering of nodes is based on level order traversal of trees. i.e. i will be n-1 for root node and so on.

Input

First line contains t, number of test cases. In each test case first line contain n and m, n is the number of nodes of **old password tree** and m is the number of nodes of **restricted password tree**. Next line contains n space separated numbers(value of nodes of **old password tree**). Next n lines contains expression like A B, in ith line A and B is the left and right child of node with index i(0 based). If there is no right or left child of a node -1 will be provided in place of any index. Next line contains m space separated numbers(value of nodes of **restricted password tree**).Then this tree is also described as the tree above.

Output

One integer. Minimum diff as per the above formula between **old tree** and **new tree** OR -1 if no such tree is possible.

Constraints

1 <= T <= 50
0 <= node_value <= 9
1 <= n <= 18

Note:
1. New tree can be same as old tree.
2. If no such tree is possible print -1
3. All trees in the problem are Binary Trees

Example

Input :
1
3 3
5 4 8
1 2
-1 -1
-1 -1
5 1 8
1 2
-1 -1
-1 -1

Output :
63

Explanation :

According to the rule specified above the most feasible tree is rooted at node with value 4 and it's left child being 8 and right child being 5.

So the minimum diff is 63.