



Tree Edit Distance

locked

by PoojanThaker

Problem

Submissions

Leaderboard

Discussions

You are given two binary trees A and B. Each node in both the trees is assigned an integral value. An operation can be of three types:

1. **Edit** Operation : Change the value of the node.
2. **Delete** Operation : Remove the node from the tree. (Only leaf nodes can be deleted)
3. **Add** Operation : Add a node to the tree. (Only leaf nodes can be added)

Find number of operations of all 3 types required to make tree A identical to Tree B such that the **total number of operations is minimum**. Note that tree A is identical to tree B if A and B have the same structure and same values of the corresponding nodes.

Note: You are only allowed to do operations on Tree A.

Input Format

The first line denotes t , the number of test cases.

The first line of a test case denotes N_A , the number of nodes in the tree A.

The next N_A lines are in the format **a b c** where a is the node, b is the left child of the node and c is the right child of the node.

The next line contains N_A space separated integers where i^{th} integer denotes value of the i^{th} node.

The next line denotes N_B , the number of nodes in the tree B.

The next N_B lines are in the format **a b c** where a is the node, b is the left child of the node and c is the right child of the node.

The next line contains N_B space separated integers where i^{th} integer denotes value of the i^{th} node.

Note 1 : Both the trees are rooted at node 1.

Note 2 : A NULL node is indicated by a value of -1.

Constraints

$t \leq 10$, Value at each node ≤ 1000000

For 2 points : Only edit operations are required to convert A to B, $n_a \leq 100$, $n_b \leq 100$

For next 2 points : Graph is linear, $n_a \leq 1000$, $n_b \leq 1000$

For next 3 points : $n_a \leq 1000$, $n_b \leq 1000$

For next 3 points : $n_a \leq 100000$, $n_b \leq 100000$

Output Format

Answer each test case on a separate line.

For a test case, print the 3 space separated integers. The number of **add** operations followed by the number of **delete** operations followed by the number of **edit** operations.

Sample Input 0

```

1
4
1 2 3
2 4 -1
3 -1 -1
4 -1 -1
1 3 4 5
4
1 2 3
2 4 -1
3 -1 -1
4 -1 -1
1 2 4 5

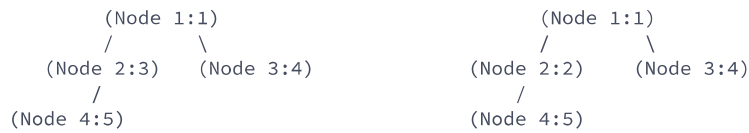
```

Sample Output 0

```
0 0 1
```

Explanation 0

The trees given are -



Note : (Node i:v) denotes node with index i and value v.

Value of node corresponding to node 2 of Tree B is edited from 3 to 2.

Sample Input 1

```

1
4
1 2 4
2 3 -1
3 -1 -1
4 -1 -1
3 3 19 4
5
1 5 4
2 -1 3
3 -1 -1
4 -1 2
5 -1 -1
3 11 9 4 2

```

Sample Output 1

```
2 1 1
```

Explanation 1

The trees given are-



Note : (Node i:v) denotes node with index i and value v.

Edit operation at corresponding Node of Node 5 of Tree B.

Add operations for adding Node 2 and Node 3 of Tree B to Tree A.

Delete operation for deleting Node 3 of Tree A.

[f](#) [t](#) [in](#)

Submissions: 0

Max Score: 10

Difficulty: Medium



Rate This Challenge:

☆☆☆☆☆

[More](#)

Admin Options

[Edit Challenge](#)[View Submissions](#)

Current Buffer (saved locally, editable)  

C  

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5
6 int main() {
7
8     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
9     return 0;
10 }
11
```

Line: 1 Col: 1

[Upload Code as File](#) ☐ Test against custom input

Run Code

Submit Code