Two Permutations

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given two permutations p and q of length n. You can do the following operation with the first permutation zero or more times:

Choose $1 \le i < j \le n$ such that $p_i < p_j$ and swap p_i with p_j.

You need to determine if it is possible to make p equal to q. If it is possible, you need to find any suitable sequence of operations.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

The first line of each test case contains an integer n ($1 \le n \le 2000$).

The second line contains n different integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$).

The third line contains n different integers $q_1, q_2, \ldots, q_n$ ($1 \le q_i \le n$).

The given p and q are both permutations of $\{1, 2, \ldots, n\}$.

The sum of n over all test cases does not exceed 2000.

Output

For each test case, determine if there exists a way to make p equal to q.

If that's the case, print "`YES`" on the first line. On the next line, print an integer k, the number of operations you wish to perform. On the next lines, you should print the operations.

To describe an operation that swaps p_i with p_j ($1 \le i < j \le n$), print a line formatted as "i j".

Otherwise, print "`NO`" on a single line.

In case there are several suitable answers, print any one of them.

Example

standard input	*standard output*
6 2 1 2 2 1 3 1 2 3 3 2 1 5 3 5 1 2 4 4 5 1 3 2 6 2 3 6 4 1 5 3 4 5 6 2 1 6 2 3 6 4 1 5 5 3 6 4 2 1 1 1 1	YES 1 1 2 YES 3 1 2 1 3 2 3 YES 2 1 5 4 5 NO YES 6 1 2 1 4 1 6 2 4 4 6 5 6 YES 0

Note

In the second test case, p is 1 2 3. One of the possible sequences of swaps is: "1 2", "1 3", "2 3".

After the first swap, p becomes 2 1 3; after the second swap, it becomes 3 1 2; and after the third, 3 2 1.