Lets Be An Anagrammatist

Do you know what is an anagram? ***An anagram is a rearrangement of letters of one word to form another word.*** For example: one of the anagram of the word “CODEMASK” can be “DEMOCSAK”. So, we can find different kinds of anagram of a word.

Now, you are given two array S & T. You have to find a **lexicographically smallest** contiguous subsequence of S which is an anagram of array T.

Between two sequence A & B, where length(A) == length(B), A will be lexicographically smaller than B if we can find an index i (1<= i <= length(A)) where A[i] < B[i] and for all j, A[j] = B[j] where 1 <= j < i.

Input:

The first line of the input is the number of the test cases Ts.

Each test case contains three lines. The first lines contains N & M, N is the length of array S & M is the length of array T.

Next line contains N integers of array S. Then another lines follows contains M integers describing array T.

Constraints:

1 <= Ts <= 20

1 <= N,M <= 200000

1 <= S[i], T[i] <= 200000

Output:

First you need to print the case number. Then on the same line, you have to print the index (1 based) of the lexicographically smallest contiguous subsequence of S which is an anagram of T. If there is more than one answer, you need to print the smallest index. If you can't find any anagram of the T in S, just print 0.

Sample Input:

```
2
4 3
1 3 2 4
1 2 3
5 3
3 2 1 4 10
1 2 4
```

Sample Output:

```
Case 1: 1
```

Case 2: 2