Old computer (old)

The year is 2341. In the last few centuries the humanity has been exponentially developing all aspects in science.

For example, during the Great Mathematical Conference it was announced, that the transitivity of equality is no longer true. And so, from a=b and b=c, it no longer means a=c. Shocking isn't it?

Since you found the old computer of your great^3 grandfather, you are wondering if it is possible to solve some of today's trivial tasks on it. After you have already solved the Longest Common Subsequence* for large strings on your Quantum Core Computer 21 in O(1), you are wondering if you will be able to solve the same task on the old computer, assuming it might not be possible in O(1). Since in the future the data is more compact, you only know the indices, in which the strings match, but not the whole strings.

Task

Your task is to create a function LcsLength, that takes 5 parameters. First two are integer numbers n1 and n2 – the amount of symbols in the first and second strings, respectively. The third is an integer number m – the amount of pairs of indices, in which the strings match. The last two parameters are arrays a and b, with m elements each. For every i, (0<=i<m) the symbols with index a_i (1<=a_i<=n1), from the first string and the symbol with index b_i (1<=b_i<=n2), from the second string match.

The function should return the maximum length of the longest common subsequence, that satisfies the described requirements. Keep in mind, that the transitivity of equality is a thing of the past!

Example

LcsLength(5, 5, 5, (1, 2, 3, 4, 5), (2, 1, 3, 5, 5)) = 3

For example the strings "ABCDD" and "BACID" match the requirements and "ACD" is one of the possible longest common subsequences for the strings.

Note that the first string could also be "ABCDE" as D=D and E=D since transitivity is missing. Eh, don't judge the wisdom of the future. If the second

string is for example “BACDD” (and the first is again “ABCDD”), LCS won’t be “ACDD”, because without transitivity it is not guaranteed that all D-s are equal.

Subtasks

Subtask 1 (15 points): n1,n2<=20, m<=20

Subtask 2 (15 points): n1,n2<=2,000, m<=100,000

Subtask 3 (10 points): n1,n2<=2,000, m<=4,000,000

Subtask 4 (30 points): n1,n2<=100,000, m<=50,000

Subtask 5 (30 points): n1,n2<=50,000,000, m<=100,000

Longest Common Subsequence* - A common subsequence of A and B, is such string C, that after deleting some symbols from A and some symbols from B, A=B=C. The longest common subsequence is the longest such string C.