Statement problem scmax

Let A be an array of numbers. Find the longest increasing subsequence of the array. In other words, remove as little numbers such that the remaining array becomes an increasing array, and print the number of numbers left.

Just kidding, who would really need this in real life?

We are interested is something deeper.

Every number a has a preferred number $T_a \leq a$ such that it is allowed that T_a can come after a in the longest increasing subsequence, if and only if the two numbers are adjacent.

Find the longest increasing subsequence, considering the new definition. An array is strictly increasing if, for each pair of consecutive numbers A_i and A_{i+1}, either $A_i \leq A_{i+1}$, or $T_{A_i} = A_{i+1}$.

Input

The input will be read from *stdin*, which contains on the first line the number of tests Q. For every test, the first line will contain the size of the array N, on the second line, N numbers between 1 and N, representing the array A and on the third and last line, N numbers between 1 and N, the i^{th} number representing T_i.

Output

The output will be written to stdout, and it has Q lines, the i^{th} line containing one number: the size of the longest increasing subsequence of the array, considering the new rules for the i^{th} test.

Restrictions

- $1 \leq Q \leq 5$
- $1 \leq A_i \leq N \leq 50000$
- For 10 points: $N \leq 14$
- For another 10 points: $N \leq 100$
- For another 10 points: $T_i = i \ \forall \ i \in [1, N]$
- For another 10 points: $1 \leq A_i \leq 3$

- For another 10 points: $\forall\ a$ there are at most three numbers b such that $T_b = a$

Example

stdin	stdout
2 10 4 1 6 1 8 10 10 1 9 1 3 6 3 2 1 9 8 1 5 10 8 8 1 8 6 4 8 5 7 7 3 2 8 8 4 5 6	5 6

Explanation

In the first test, the longest increasing subsequence is 4, 6, 8, 10, 10. In the second test, the longest increasing subsequence is 1, 8, 6, 4, 5, 7.