Problem G. Couleur

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

DreamGrid has an array of n integers. On this array he can perform the following operation: choose an element that was not previously chosen and mark it as unavailable. DreamGrid would like to perform exactly n operations until all the elements are marked.

DreamGrid defines the cost of a subarray as the number of inversions in the subarray. Before performing an operation, DreamGrid would like to know the maximum cost of a subarray that doesn't contain any unavailable elements.

Recall that a subarray $a_l, a_{l+1}, \ldots, a_{r-1}, a_r$ is a **contiguous** subpart of the original array where $1 \le l \le r \le n$. An inversion in a subarray $a_l, a_{l+1}, \ldots, a_{r-1}, a_r$ is a pair of indices (i, j) $(l \le i < j \le r)$ such that the inequality $a_i > a_j$ holds.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains a single integer n $(1 \le n \le 10^5)$ – the length of the array.

The second line contains the n values of the array $a_1, a_2, ..., a_n$ $(1 \le a_i \le n)$.

The third line contains a permutation $p_1, p_2, \ldots, p_n$, representing the indices of the elements chosen for the operations in order.

Note that the permutation is encrypted and you can get the real permutation using the following method: Let z_i be the answer before the i-th operation. The actual index of the i-th operation is $p_i \oplus z_i$ where $\oplus$ is bitwise exclusive or operator.

It is guaranteed that the sum of all n does not exceed 10^6.

Output

For each test case, output n integers $z_1, z_2, \ldots, z_n$ in a single line seperated by one space, where z_i is the answer before the i-th operation.

Please, DO NOT output extra spaces at the end of each line, or your answer may be considered incorrect!

Example

standard input	standard output
3 5 4 3 1 1 1 5 4 5 3 1 10 9 7 1 4 7 8 5 7 4 8 21 8 15 5 9 2 4 5 10 6 15 4 8 8 1 12 1 10 14 7 14 2 9 13 10 3 37 19 23 15 7 2 10 15 2 13 4 5 8 7 10	7 0 0 0 0 20 11 7 2 0 0 0 0 0 0 42 31 21 14 14 4 1 1 1 0 0 0 0 0 0

Note

The decoded permutation of each test case is $\{2, 4, 5, 3, 1\}$, $\{1, 3, 8, 7, 9, 2, 4, 5, 10, 6\}$ and $\{15, 12, 2, 1, 9, 6, 11, 14, 3, 13, 4, 5, 8, 7, 10\}$