Kick Start 2022 - Round H

Level Design

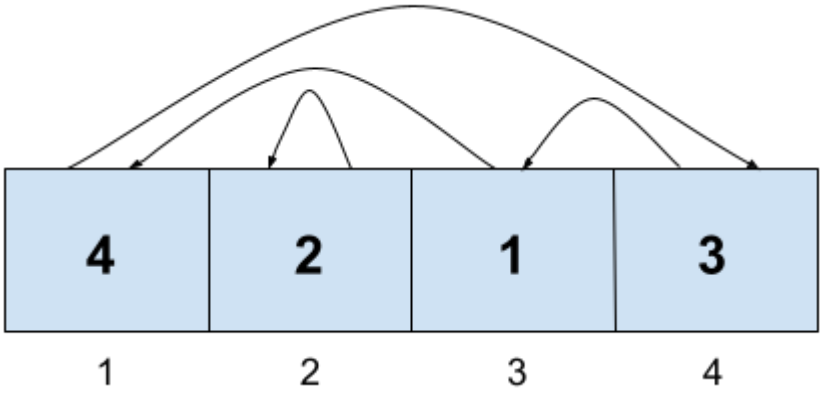
Problem

A permutation cycle in a permutation C is a sequence of integers $(a_1, a_2, \ldots, a_k)$ such that the following hold:

- $a_i \in C$ for all i, and are distinct.
- For each $i \in \{1, 2, \ldots, k-1\}$: $C[a_i] = a_{i+1}$, and $C[a_k] = a_1$.

A permutation cycle of length k is called a k-cycle.

For example, the permutation $C = [4, 2, 1, 3]$ has two cycles: the 3-cycle $(4, 3, 1)$, and the 1-cycle (2). $(4, 3, 1)$ is a cycle because $C[4] = 3$, $C[3] = 1$, and $C[1] = 4$.



Grace loves permutation cycles, so Charles decides to design an $\mathbf{N}$-level game to challenge her.

At the start of the game, the player is given an $\mathbf{N}$-length permutation $\mathbf{P}$ of integers from 1 through $\mathbf{N}$. The levels in the game are numbered from 1 to $\mathbf{N}$. At each level, the player starts with the *given* permutation, and is allowed to make modifications to it by swapping any two elements in it (multiple swaps allowed). To clear the k-th level in the game, the player is required to find the *minimum* number of swaps using which a k-cycle can be created in the permutation. The player can progress to the $(k+1)$-th level only after clearing the k-th level.

Grace finds the game a bit challenging, but wants to win at any cost. She needs your help! Formally, for each level k, you need to find the minimum number of swaps using which a k-cycle can be created in the permutation.

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow.
The first line of each test case contains an integer $\mathbf{N}$: the length of the permutation.
The next line contains $\mathbf{N}$ integers $\mathbf{P_1}$, $\mathbf{P_2}$, ..., $\mathbf{P_N}$, where the i-th integer represents the i-th element in the permutation $\mathbf{P}$.

Output

For each test case, output one line containing `Case #`x`:` $S_1, S_2, \ldots, S_N$, where x is the test case number (starting from 1), and S_i is the solution for the i-th level in the game, that is, the minimum number of swaps needed to create an i-cycle in the permutation.

Limits

Time limit: 20 seconds.
Memory limit: 1 GB.
$1 \leq \mathbf{T} \leq 100$.
$1 \leq \mathbf{P_i} \leq \mathbf{N}$, for all i.
All $\mathbf{P_i}$ are distinct.

Test Set 1

$1 \leq \mathbf{N} \leq 10^3$.

Test Set 2

$1 \leq \mathbf{N} \leq 10^5$.

Sample

Sample Input

```
2
3
1 2 3
4
4 2 1 3
```

Sample Output

```
Case #1: 0 1 2
Case #2: 0 1 0 1
```

In Sample Case #1, there are three 1-cycles in the given permutation. So, the first level can be cleared with zero swaps. To clear the second level, we can swap the first two elements to get the permutation $[2, 1, 3]$, which contains the 2-cycle $(2, 1)$. To clear the third level, we can swap the first two elements, followed by the second and third elements to get the permutation $[2, 3, 1]$, which contains the 3-cycle $(2, 3, 1)$.

In Sample Case #2, as explained earlier, the permutation has the 1-cycle (2). So, zero swaps are needed to clear the first level. To clear the second level, we can swap the last two elements to get the permutation $[4, 2, 3, 1]$, which contains the 2-cycle $(4, 1)$. Since the permutation also has the 3-cycle $(4, 3, 1)$, the third level can also be cleared using zero swaps. To clear the fourth level, we can swap the second and the fourth elements to get the permutation $[4, 3, 1, 2]$, which contains the 4-cycle $(4, 2, 3, 1)$.