

GoroSort

Problem

Goro has 4 arms. Goro is very strong. You don't mess with Goro. Goro needs to sort an array of **N** different integers. Algorithms are not Goro's strength; strength is Goro's strength. Goro's plan is to use the fingers on two of his hands to hold down several elements of the array and hit the table with his third and fourth fists as hard as possible. This will make the unsecured elements of the array fly up into the air, get shuffled randomly, and fall back down into the empty array locations.

Goro wants to sort the array as quickly as possible. How many hits will it take Goro to sort the given array, on average, if he acts intelligently when choosing which elements of the array to hold down before each hit of the table? Goro has an infinite number of fingers on the two hands he uses to hold down the array.

More precisely, before each hit, Goro may choose any subset of the elements of the array to freeze in place. He may choose differently depending on the outcomes of previous hits. Each hit permutes the unfrozen elements uniformly at random. Each permutation is equally likely.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each one will consist of two lines. The first line will give the number **N**. The second line will list the **N** elements of the array in their initial order.

Output

For each test case, output one line containing "Case #**x**: **y**", where **x** is the case number (starting from 1) and **y** is the expected number of hit-the-table operations when following the best hold-down strategy. Answers with an absolute or relative error of at most 10^{-6} will be considered correct.

Limits

$1 \leq T \leq 100$;

The second line of each test case will contain a permutation of the **N** smallest positive integers. Memory limit: 1GB.

Small dataset (Test set 1 - Visible)

$1 \leq N \leq 10$;

Time limit: 30 seconds.

Large dataset (Test set 2 - Hidden)

$1 \leq N \leq 1000$;

Time limit: 60 seconds.

Sample

Sample Input

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

Sample Output

```
Case #1: 2.000000
Case #2: 2.000000
Case #3: 4.000000
```

Explanation

In test case #3, one possible strategy is to hold down the two leftmost elements first. Elements 3 and 4 will be free to move. After a table hit, they will land in the correct order [3, 4] with probability $1/2$ and in the wrong order [4, 3] with probability $1/2$. Therefore, on average it will take 2 hits to arrange them in the correct order. After that, Goro can hold down elements 3 and 4 and hit the table until 1 and 2 land in the correct order, which will take another 2 hits, on average. The total is then $2 + 2 = 4$ hits.