Sorting

Description

For a given permutation of $1 \ldots n$, noting it as P, you need to insert the number in P into a set S one by one, i.e. $P_1, P_2, \ldots, P_{n-1}, P_n$.

After the insertion of each number, you need to find the largest number in S that is smaller than the inserted number and the smallest number in S that is larger than the inserted number.

Initially, two integers $0, n+1$ has **already been inserted into** S so that the results always exist.

Input

The input contains multiple test cases. The first line of the input contains an integer T, indicating the number of test cases.

For each test case, the first line contains a positive integer n. The second line contains n integers, separated by spaces, the i-th integer of which denotes P_i.

Output

For each test case, print n lines, each line contains two integers separated by a space, the i-th line of which denotes the result required after the insertion of the i-th number.

The first integer of a result represents the largest number in S that is smaller than the inserted number and the second one represents the smallest number in S that is larger than the inserted numbser.

Sample Input/Output

Input

```
2
5
1 5 2 4 3
2
1 2
```

Output

```
0 6
1 6
1 5
2 5
2 4
0 3
1 3
```

Constraint

$1 \leq n \leq 10^5$.

It is guaranteed that P is a permutation of $1 \dots n$.

Hint

To find all the results quickly enough, you may think about reverting the whole process.