E. MEX and Increments

time limit per test

2 seconds

memory limit per test

256 megabytes

input

standard input

output

standard output

Dmitry has an array of n non-negative integers a1,a2,…,an.

In one operation, Dmitry can choose any index j (1≤j≤n) and increase the value of the element aj by 1. He can choose the same index j multiple times.

For each i from 0 to n, determine whether Dmitry can make the MEX of the array equal to exactly i. If it is possible, then determine the minimum number of operations to do it.

The MEX of the array is equal to the minimum non-negative integer that is not in the array. For example, the MEX of the array [3,1,0] is equal to 2, and the array [3,3,1,4] is equal to 0.

**Input**

The first line of input data contains a single integer t (1≤t≤104) — the number of test cases in the input.

The descriptions of the test cases follow.

The first line of the description of each test case contains a single integer n (1≤n≤2⋅105) — the length of the array a.

The second line of the description of each test case contains n integers a1,a2,…,an (0≤ai≤n) — elements of the array a.

It is guaranteed that the sum of the values n over all test cases in the test does not exceed 2⋅105.

**Output**

For each test case, output n+1 integer — i-th number is equal to the minimum number of operations for which you can make the array MEX equal to i (0≤i≤n), or -1 if this cannot be done.

**Example**

**input**

**Copy**

5

3

0 1 3

7

0 1 2 3 4 3 2

4

3 0 0 0

7

4 6 2 3 5 0 5

5

4 0 1 0 4

**output**

**Copy**

1 1 0 -1

1 1 2 2 1 0 2 6

3 0 1 4 3

1 0 -1 -1 -1 -1 -1 -1

2 1 0 2 -1 -1

**Note**

In the first set of example inputs, n=3:

* to get MEX=0, it is enough to perform one increment: a1++;
* to get MEX=1, it is enough to perform one increment: a2++;
* MEX=2 for a given array, so there is no need to perform increments;
* it is impossible to get MEX=3 by performing increments.