C. Trinity

time limit per test: 2 seconds
memory limit per test: 256 megabytes

You are given an array a of n elements $a_1, a_2, \ldots, a_n$.

You can perform the following operation any number (possibly 0) of times:

- Choose two integers i and j, where $1 \le i, j \le n$, and assign $a_i := a_j$.

Find the minimum number of operations required to make the array a satisfy the condition:

- For every pairwise distinct triplet of indices (x, y, z) ($1 \le x, y, z \le n$, $x \ne y$, $y \ne z$, $x \ne z$), there exists a non-degenerate triangle with side lengths a_x, a_y and a_z, i.e. $a_x + a_y > a_z$, $a_y + a_z > a_x$ and $a_z + a_x > a_y$.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n ($3 \le n \le 2 \cdot 10^5$) — the number of elements in the array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the array a.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer — the minimum number of operations required.

Example

input Copy

```
4
7
1 2 3 4 5 6 7
3
1 3 2
3
4 5 3
15
9 3 8 1 6 5 3 8 2 1 4 2 9 4 7
```

output Copy

```
3
1
0
8
```

Note

In the first test case, one of the possible series of operations would be:

- Assign $a_1 := a_4 = 4$. The array will become $[4, 2, 3, 4, 5, 6, 7]$.
- Assign $a_2 := a_5 = 5$. The array will become $[4, 5, 3, 4, 5, 6, 7]$.
- Assign $a_7 := a_1 = 4$. The array will become $[4, 5, 3, 4, 5, 6, 4]$.

It can be proven that any triplet of elements with pairwise distinct indices in the final array forms a non-degenerate triangle, and there is no possible answer using less than 3 operations.

In the second test case, we can assign $a_1 := a_2 = 3$ to make the array $a = [3, 3, 2]$.

In the third test case, since 3, 4 and 5 are valid side lengths of a triangle, we don't need to perform any operation to the array.