A. Remove Smallest

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given the array a consisting of n positive (greater than zero) integers.

In one move, you can choose two indices i and j ($i \neq j$) such that the absolute difference between a_i and a_j is no more than one ($|a_i - a_j| \leq 1$) and remove the smallest of these two elements. If two elements are equal, you can remove any of them (but exactly one).

Your task is to find if it is possible to obtain the array consisting of **only one element** using several (possibly, zero) such moves or not.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \leq t \leq 1000$) — the number of test cases. Then t test cases follow.

The first line of the test case contains one integer n ($1 \leq n \leq 50$) — the length of a. The second line of the test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 100$), where a_i is the i-th element of a.

Output

For each test case, print the answer: "YES" if it is possible to obtain the array consisting of **only one element** using several (possibly, zero) moves described in the problem statement, or "NO" otherwise.

Example

input Copy

```
5
3
1 2 2
4
5 5 5 5
3
1 2 4
4
1 3 4 4
1
100
```

output Copy

```
YES
YES
NO
NO
YES
```

Note

In the first test case of the example, we can perform the following sequence of moves:

- choose $i = 1$ and $j = 3$ and remove a_i (so a becomes $[2; 2]$);
- choose $i = 1$ and $j = 2$ and remove a_j (so a becomes $[2]$).

In the second test case of the example, we can choose any possible i and j any move and it doesn't matter which element we remove.

In the third test case of the example, there is no way to get rid of 2 and 4.
