

Containers

We are given n containers, where $1 \leq n \leq 4$. At the beginning all of them are full of water. The liter capacity of the i -th container is a natural number o_i satisfying inequalities $1 \leq o_i \leq 49$.

Three kinds of moves can be made:

1. Pouring the whole content of one container into another. This move can be made unless there is too little room in the second container.
2. Filling up one container with part of the water from another one.
3. Pouring away the whole content of one container into a drain.

Task

Write a program that for each test case:

- Reads the number of containers n , the capacity of each container and the requested final amount of water in each container.
- Verifies, whether there exist a series of moves which leads to the requested final situation, and if there is one, the program computes the minimal number of moves leading to the requested situation,
- Writes the result. The result should be the minimal number of moves leading to the requested final situation, or one word "NO" if there is no such a sequence of moves.

Input

One integer in the first line, stating the number of test cases, followed by a blank line. There will be not more than 20 tests.

For each test case, at the first line, one positive integer n is written, $n \leq 4$, this is the number of containers. There are n positive integers written in the second line. These are the capacities of the containers (the i -th integer o_i denotes the capacity of the i -th container, $1 \leq o_i \leq 49$). In the third line there are written n numbers. These are the requested final volumes of water in the containers (the i -th integer w_i denotes the requested final volume of water in the i -th container, $0 \leq w_i \leq o_i$). All integers in the second and the third line are separated by single spaces.

The test cases will be separated by a single blank line.

Output

For each test case : write one integer - the minimal number of moves which lead to the requested final situation or write only one word "NO" if it is not possible to reach the requested final situation making only allowed moves.

Example

Input:

2

3

3 5 5
0 0 4

2
20 25
10 16

Output:

6
NO