Problem # 2: Bear and Cavalry

(Input File: in2.txt, Output: Console Output)

Would you want to fight against bears riding horses? Me neither.

Limak is a grizzly bear. He is a general of the dreadful army of Bearland. The most important part of an army is the cavalry of course.

The cavalry of Bearland consists of **N** warriors and **N** horses, both numbered **1** through **N**. Limak knows the strength of each warrior $W_1, W_2, ..., W_N$ and the strength of each horse $H_1, H_2, ..., H_N$.

A warrior together with his horse is called a unit. The strength of a unit is equal to the multiplied strengths of a warrior and a horse.

General Limak must assign all horses to warriors, one horse per warrior. The cavalry will consist of **N** units then.

The first warrior (the one with the strength W_1) is called Bravebeart. He is always the first to charge the enemy. Limak decided that Bravebeart deserves some respect and his unit must be the strongest one, with no ties allowed. But is it possible?

Help Limak and check whether there is an assignment of horses to warriors for which the Bravebeart's unit is strictly stronger than any other unit. Print **"YES"** or **"NO"**.

Input format

You are given multiple test cases.

The first line of input contains a single integer **T**, denoting the number of test cases.

For each test case the first line contains a single integer **N**.

The second line contains **N** integer numbers $W_1, W_2, ..., W_N$, denoting the strengths of warriors. The first of the numbers is the strength of Bravebeart.

The third line contains **N** integers $H_1, H_2, ..., H_N$, denoting the strengths of the horses.

Output format

For each test case find the answer and print it in a separate line.

Print **"YES"** (without the quotes) if there is an assignment where the strength of the Bravebeart's unit is strictly greater than the strength of any other unit. Otherwise, print **"NO"** (without the quotes).

Constraints

- $1 \le T \le 50$
- $2 \le N \le 42$
- $1 \le W_i, H_i \le 1000$

SAMPLE INPUT

```
2
6
12 9 7 1 20 10
3 6 4 7 5 5
4
```

5 17 10 1
200 400 800 600

SAMPLE OUTPUT

YES
NO