

Problem D - Supersort

In your computer science studies you have heard of many different sorting algorithms. You have studied simple binary sort, but also quick sort, merge sort, bucket sort, radix sort and many more.

Another idea is to do a combination of burstsort and quicksort. Therefore, a tree of buckets is build and then for each input string, the program walks through the tree and inserts part of the string into the right bucket. When a bucket fills up, it "bursts" and becomes a new subtree (with new buckets).

Well, enough of these algorithms. Today we want to talk about a new algorithm. It is extremely fast, but we are not sure if it works correctly yet. We will call it *Supersort*, as the so called super property shall hold after the first step of this algorithm. The super property is defined as $\min_{1 \leq i, j \leq N} (x_{i,j}) \geq 0$, where

$$x_{i,j} = \begin{cases} a_j - a_i & \text{for } 1 \leq i \leq j \leq N \\ 9001 & \text{otherwise} \end{cases}$$

Input

The first line contains a single integer N ($1 \leq N \leq 1024$). The second line contains N integers $a_1 a_2 \dots a_N$ ($1 \leq a_i \leq 1024$).

Output

Print one line of output containing **yes** if the extreme property holds for the given input, **no** otherwise.

Sample Input 1

2
1 2

Sample Output 1

yes

Sample Input 2

4
2 1 3 4

Sample Output 2

no