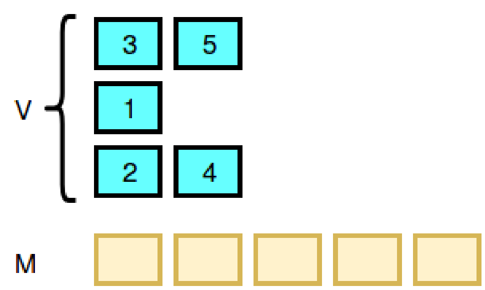
Sherlock's Array Merging Algorithm

Watson gave Sherlock a collection of arrays V. Here each V_i is an array of variable length. It is guaranteed that if you merge the arrays into one single array, you'll get an array, M, of n distinct integers in the range $[1, n]$.

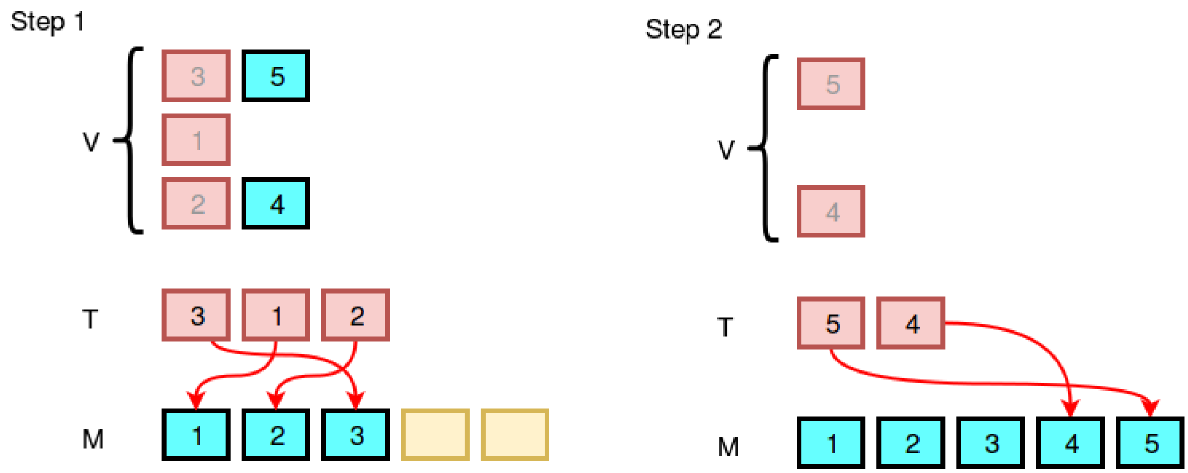
Watson asks Sherlock to merge V into a sorted array. Sherlock is new to coding, but he accepts the challenge and writes the following algorithm:

- $M \leftarrow [\,]$ (an empty array).
- $k \leftarrow$ number of arrays in the collection V.
- While there is at least one non-empty array in V:
  - $T \leftarrow [\,]$ (an empty array) and $i \leftarrow 1$.
  - While $i \leq k$:
    - If V_i is not empty:
      - Remove the first element of V_i and push it to T.
    - $i \leftarrow i + 1$.
  - While T is not empty:
    - Remove the minimum element of T and push it to M.
- Return M as the *output*.

Let's see an example. Let V be $\{[3, 5], [1], [2, 4]\}$.



The image below demonstrates how Sherlock will do the merging according to the algorithm:



Sherlock isn't sure if his algorithm is correct or not. He ran Watson's *input*, V, through his pseudocode algorithm to produce an *output*, M, that contains an array of n integers. However, Watson forgot the contents of V and only has Sherlock's M with him! Can you help Watson reverse-engineer M to get the original contents of V?

Given m, find the number of different ways to create collection V such that it produces m when given to Sherlock's algorithm as *input*. As this number can be quite large, print it modulo $10^9 + 7$.

Notes:

- Two collections of arrays are *different* if one of the following is *true*:
  - Their sizes are different.
  - Their sizes are the same but at least one array is present in one collection but not in the other.
- Two arrays, A and B, are different if one of the following is *true*:
  - Their sizes are different.
  - Their sizes are the same, but there exists an index i such that $a_i \neq b_i$.

Input Format

The first line contains an integer, n, denoting the size of array M.
The second line contains n space-separated integers describing the respective values of $m_0, m_1, \ldots, m_{n-1}$.

Constraints

- $1 \leq n \leq 1200$
- $1 \leq m_i \leq n$

Output Format

Print the number of different ways to create collection V, modulo $10^9 + 7$.

Sample Input 0

```
3
```

```
1 2 3
```

Sample Output 0

```
4
```

Explanation 0

There are four distinct possible collections:

1. $V = \{[1, 2, 3]\}$
2. $V = \{[1], [2], [3]\}$
3. $V = \{[1, 3], [2]\}$
4. $V = \{[1], [2, 3]\}$.

Thus, we print the result of $4 \bmod (10^9 + 7) = 4$ as our answer.

Sample Input 1

```
2
2 1
```

Sample Output 1

```
1
```

Explanation 1

The only distinct possible collection is $V = \{[2, 1]\}$, so we print the result of $1 \bmod (10^9 + 7) = 1$ as our answer.