

Problem A. Number Of LIS

A. Number Of LIS

Time Limit: 5s

Memory Limit: 256 megabytes

Input: standart input

Ouptut: standart output

Cho dãy số $A_1, A_2, A_3, \dots, A_n$. Hãy đếm số lượng dãy con tăng dài nhất của dãy số trên.

Một dãy con độ dài k của dãy A được xác định bởi một bộ chỉ số $(u_1 < u_2 < u_3 < \dots < u_k)$ ($1 \leq u_i \leq n$). Hai dãy con $(u_1, u_2, u_3, \dots, u_k)$ và $(v_1, v_2, v_3, \dots, v_t)$ được gọi là khác nhau nếu $k \neq t$ hoặc tồn tại một vị trí i sao cho $u_i \neq v_i$.

Input

- Dòng đầu tiên ghi số nguyên dương T là số test ($T \leq 100$)
- Với mỗi test dòng đầu tiên là số nguyên dương n là số phần tử của mảng A . Dòng tiếp theo ghi n số nguyên mô tả dãy A ($A_i \leq 10^9$)

Output

- In ra T dòng. Dòng thứ i in ra một số duy nhất là kết quả của test thứ i theo module $10^9 + 7$

Giới hạn:

- Dữ liệu đảm bảo $\sum n$ của tất cả các test không vượt quá $2 * 10^5$

Ví dụ:

Input	Output
3	8
6	1
1 1 2 2 3 3	4
5	
1 2 3 4 5	
4	
10 6 3 1	

Problem A. Number Of LIS

Number Of LIS

Time Limit: 5s

Memory Limit: 256 megabytes

Input: standart input

Ouput: standart output

Given a sequence of number $A_1, A_2, A_3, \dots, A_n$. Count the number of longest increasing subsequences.

A sub-sequence of length k of A is defined by an index set $(u_1 < u_2 < u_3 < \dots < u_k)$ ($1 \leq u_i \leq n$). The two sub sequences $(u_1, u_2, u_3, \dots, u_k)$ and $(v_1, v_2, v_3, \dots, v_t)$ are called different if $k \neq t$ or exists a position i such that $u_i \neq v_i$.

Input

- The first line contains a positive integer T , which indicates the number of test cases ($T \leq 100$).
- For each test, the first line is a positive integer n , which is the number of elements in array A . The next line contains n integers describing the sequence A ($A_i \leq 10^9$).

Output

- Print T lines. On the i^{th} line is a unique number that is the result of the i^{th} test according to the $10^9 + 7$ module.

Constraints:

- Data guarantee $\sum n$ of all test cases do not exceed $2 * 10^5$

Examples:

Input	Output
3	8
6	1
1 1 2 2 3 3	4
5	
1 2 3 4 5	
4	
10 6 3 1	