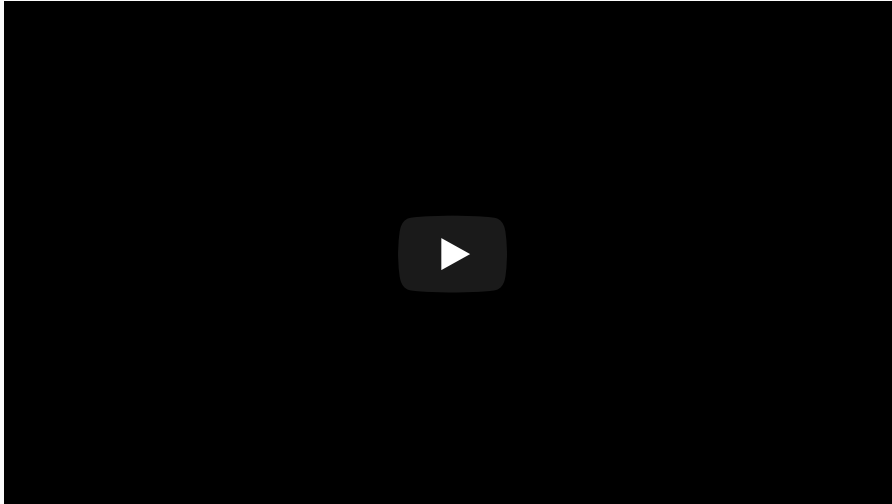


## An Introduction to the Longest Increasing Subsequence Problem

The task is to find the length of the longest subsequence in a given array of integers such that all elements of the subsequence are sorted in strictly ascending order. This is called the Longest Increasing Subsequence (LIS) problem.

For example, the length of the LIS for `[15, 27, 14, 38, 26, 55, 46, 65, 85]` is **6** since the longest increasing subsequence is `[15, 27, 38, 55, 65, 85]`.

Here's a great YouTube video of a lecture from MIT's Open-CourseWare covering the topic.



This is one approach which solves this in quadratic time using dynamic programming. A more efficient algorithm which solves the problem in  $O(n \log n)$  time is [available here](#).

Given a sequence of integers, find the length of its longest strictly increasing subsequence.

### Function Description

Complete the *longestIncreasingSubsequence* function in the editor below. It should return an integer that denotes the array's LIS.

`longestIncreasingSubsequence` has the following parameter(s):

- *arr*: an unordered array of integers

### Input Format

The first line contains a single integer *n*, the number of elements in *arr*.  
Each of the next *n* lines contains an integer, *arr[i]*

### Constraints

- $1 \leq n \leq 10^6$
- $1 \leq arr[i] \leq 10^5$

### Output Format

Print a single line containing a single integer denoting the length of the longest increasing subsequence.

### Sample Input 0

```
5
2
7
4
3
8
```

### Sample Output 0

3

### Explanation 0

In the array  $arr = [2, 7, 4, 3, 8]$ , the longest increasing subsequence is  $[2, 7, 8]$ . It has a length of **3**.

### Sample Input 1

6  
2  
4  
3  
7  
4  
5

### Sample Output 1

4

### Explanation 1

The LIS of  $arr = [2, 4, 3, 7, 4, 5]$  is  $[2, 3, 4, 5]$ .