

# The Longest Increasing Subsequence

## An Introduction to the Longest Increasing Subsequence Problems

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 ascending order. For example, the length of the LIS for { 15, 27, 14, 38, 26, 55, 46, 65, 85 } is 6 and 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.

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

In this challenge you simply have to find the length of the longest strictly increasing sub-sequence of the given sequence.

### Input Format

In the first line of input, there is a single number  $N$ .  
In the next  $N$  lines input the value of  $a[i]$ .

### Constraints

$$1 \leq N \leq 10^6$$
$$1 \leq a[i] \leq 10^5$$

### Output Format

In a single line, output the length of the longest increasing sub-sequence.

### Sample Input

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

## Sample Output

3

## Explanation

{2,7,8} is the longest increasing sub-sequence, hence the answer is 3 (the length of this sub-sequence).