

The Longest Increasing Subsequence





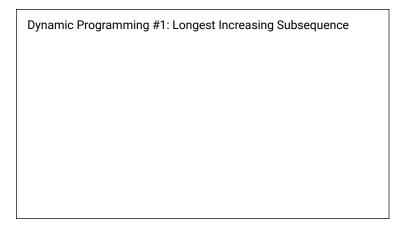


An Introduction to the Longest Increasing Subsequence Problems

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-Coursware, 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

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

Output Format

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

Sample Input

Sample Output

3

Explanation

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

f in

Submissions: 12291

Max Score: 60

Difficulty: Advanced

Rate This Challenge:

ななななな

```
Current Buffer (saved locally, editable) & 🗗
                                                                                            Java 8
                                                                                                                              *
1 ▼ import java.io.*;
2
3 ▼ public class Solution {
4
5 1
        public static void main(String[] args) throws IOException{
6
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7
8
            int N = Integer.parseInt(br.readLine());
9
10 🔻
             int[] arr = new int[N];
11
12 ▼
            for(int i = 0; i < N; i++){
                 arr[i] = Integer.parseInt(br.readLine());
13 ▼
14
15
            System.out.println(getLen(arr,arr.length));
16
17
18
        }
19
        public static int getLen(int[] A, int size){
20 ▼
21
            int[] tailTable = new int[size];
22 ▼
23
            int len = 0;
24
25 ▼
            tailTable[0] = A[0];
26
            len++;
27
28 ▼
            for(int i = 1; i < size; i++){</pre>
29
                 if(A[i] < tailTable[0]){</pre>
30 ▼
                     tailTable[0] = A[i];
31 ▼
32
33 ₹
                 else if(A[i] > tailTable[len - 1]){
34 ₹
                     tailTable[len] = A[i];
35
                     len++;
36
37 ▼
                 else{
38 ▼
                     int in = binarySearch(tailTable, -1, len - 1, A[i]);
39 ▼
                     if(in == 0){
40
                         continue;
41
```

```
8/29/2017
                                           The Longest Increasing Subsequence | Algorithms Question | HackerRank
                         tailTable[in] = A[i];
    42 🔻
    43
                     }
    44
                 }
    45
    46
                 return len;
    47
    48
    49 ▼
             public static int binarySearch(int arr[], int leftIndex, int rightIndex, int key){
    50
    51 ▼
                 while(rightIndex - 1 > leftIndex){
    52
                     int index = (leftIndex + rightIndex) / 2;
    53
    54
    55 ▼
                     if(arr[index] >= key){
    56
                         rightIndex = index;
    57
    58 ▼
                     else{
    59
                         leftIndex = index;
    60
    61
                 return rightIndex;
    62
             }
    63
    64
        }
                                                                                                                           Line: 1 Col: 1
```

<u>**1**</u> <u>Upload Code as File</u> ☐ Test aga

☐ Test against custom input

Run Code

Submit Code

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature