

[All Tracks](#) > [Data Structures](#) > [Trees](#) > > Problem

10



Monk watching fight

Attempted by: **844** / Accuracy: **91%** / Maximum Score: **20** / ★★★★★ 4 Votes

Tag(s): Binary Search Tree, Data Structures, Easy

PROBLEM**EDITORIAL****MY SUBMISSIONS**

Once Monk was watching a fight between an array and a tree, of being better. Tree got frustrated and converted that array into a Binary Search Tree by inserting the elements as nodes in BST, processing elements in the given order in the array. Now Monk wants to know the height of the created Binary Search Tree.

Help Monk for the same.

Note:

- 1) In Binary Search Tree, the left sub-tree contains only nodes with values less than or equal to the parent node; the right sub-tree contains only nodes with values greater than the parent node.
- 2) Binary Search Tree with one node, has height equal to 1.

Input Format :

The first line will consist of 1 integer N , denoting the number of elements in the array.
In next line, there will be N space separated integers, $A[i]$, where $1 \leq i \leq N$, denoting the elements of array.

Output Format

Print the height of the created Binary Search Tree.

Constraints: :

$$1 \leq N \leq 10^3$$

$$1 \leq A[i] \leq 10^6$$

SAMPLE INPUT

```
4
2 1 3 4
```

SAMPLE OUTPUT

```
3
```

Explanation

$N = 4$.

Insert 2. It becomes root of the tree.

Insert 1. It becomes left child of 2.

Insert 3. It becomes right child of 2.

Insert 4. It becomes right child of 3.

Final height of tree = 3.

Time Limit: 1.0 sec(s) for each input file.

Memory Limit: 256 MB

Source Limit: 1024 KB

Marking Scheme: Marks are awarded when all the testcases pass.

Allowed Languages: C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

CODE EDITOR

Enter your code or [Upload your code](#) as file.

Save

C (gcc 4.8.2)



```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello World!\n");
6     return 0;
7 }
8
```

1:1

☒ Provide custom input

COMPILE & TEST

SUBMIT

Press Ctrl-space for autocomplete suggestions.

POWERED BY code table

Your Rating:

Like

Share

0

Tweet