All Tracks › Data Structures › Stacks › › Problem

10

LIVE EVENTS

Little Monk and Balanced Parentheses

Attempted by: **986** / Accuracy: **96%** / Maximum Score: **30** / ★★★★☆ 8 Votes

Tag(s): Data Structures, Easy-Medium, Stack, approved

PROBLEM EDITORIAL MY SUBMISSIONS

Given an array of positive and negative integers, denoting different types of parentheses. The positive numbers x_i denotes opening parentheses of type x_i and negative number $-x_i$ denotes closing parentheses of type x_i.

Open parentheses must be closed by the same type of parentheses. Open parentheses must be closed in the correct order, i.e., never close an open pair before its inner pair is closed (if it has an inner pair). Thus, $[1, 2, -2, -1]$ is balanced, while $[1, 2, -1, -2]$ is not balanced.

You have to find out the length of the longest subarray that is balanced.

Input Format:

First line contains an input N $(1 \leq N \leq 2 * 10^5)$, denoting the number of parentheses. Second line contains N space separated integers. x_i $(-10^5 \leq x_i \leq 10^5, x_i \neq 0)$ denoting the i^{th} parentheses of the array.

Output Format:

Print the length of the longest subarray that is balanced.

SAMPLE INPUT

```
5
1 -1 2 3 -2
```

SAMPLE OUTPUT

```
2
```

Explanation

The longest subarray that is balanced is $(1, -1)$. $(2, 3, -2)$ is not balanced as (3) is not balanced.

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,