

ВШЭ АиСД 2021. Сбалансированные деревья

4 дек 2021, 18:43:57

старт: 29 окт 2021, 12:00:00

финиш: 9 ноя 2021, 00:00:01

длительность: 10д. 12ч.

начало: 29 окт 2021, 12:00:00

конец: 9 ноя 2021, 00:00:01

В. AVL-сбалансированность (0.25)

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

Дерево называется **AVL-сбалансированным**, если для любой его вершины высота левого и правого поддеревя для этой вершины различаются не более чем на 1.

Формат ввода

На вход программа получает последовательность целых чисел (пусть n - длина последовательности, $0 \leq n \leq 2 \cdot 10^4$; числа $\in [-10^9; 10^9]$). Последовательность завершается числом 0 (оно не входит в последовательность). Постройте дерево, соответствующее данной последовательности.

Формат вывода

Определите, является ли дерево сбалансированным, выведите слово YES или NO.

Система оценки

Группа	Баллы	Доп. ограничения	Необх. группы	Комментарий
		n		
0	1	–	–	Тесты из условия.
1	0,5	$n \leq 50$	0	
2	1	$n \leq 10^3$	0	
3	1,5	$n \leq 10^4$	0	
4	2	$n \leq 10^5$	0	Небольшая высота.
5	2	$n \leq 15000$	0 – 4	Offline-проверка

Пример

Ввод

Вывод

7 3 2 1 9 5 4 6 8 0

YES

```
1 #include <iostream>
2
3 struct TreeNode {
4     int value;
5     int height;
6     TreeNode *leftChildren;
7     TreeNode *rightChildren;
8
9     explicit TreeNode(int new_value) {
10         value = new_value;
11         height = 1;
12         leftChildren = nullptr;
13         rightChildren = nullptr;
14     }
15 };
16
17 class Tree {
18 private:
19     TreeNode *root_;
20
21     TreeNode *insertChildren(TreeNode *tree_node, int value) {
22         if (tree_node == nullptr) {
23             tree_node = new TreeNode(value);
24         } else if (value < tree_node->value) {
25             tree_node->leftChildren = insertChildren(tree_node->leftChildren, value);
26         } else if (value > tree_node->value) {
27             tree_node->rightChildren = insertChildren(tree_node->rightChildren, value);
28         }
29
30         return fixHeight(tree_node);
31     }
32
33     static int maximum(int first_argument, int second_argument) {
34         return first_argument > second_argument ? first_argument : second_argument;
35     }
36
37     static TreeNode *fixHeight(TreeNode *tree_node) {
38         int height_left_children = 0;
```

[Отправить](#)[Предыдущая](#)[Следующая](#)