

ВШЭ АиСД 2021. Деревья (noSTL)

4 дек 2021, 18:45:02

старт: 22 окт 2021, 12:00:00

финиш: 31 окт 2021, 23:59:59

длительность: 9д. 11ч.

начало: 22 окт 2021, 12:00:00

конец: 31 окт 2021, 23:59:59

D. Вывод листьев (0.25)

Ограничение времени	2 секунды
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

Для полученного дерева выведите список всех листьев (вершин, не имеющих потомков) **в порядке возрастания**.

Формат ввода

На вход программа получает последовательность натуральных чисел (пусть  $n$  - длина последовательности,  $0 \leq n \leq 11000$ ; числа  $\in [1; 10^9]$ ). Последовательность завершается числом 0 (оно не входит в последовательность), которое означает конец ввода, и добавлять его в дерево не надо.

Формат вывода

Выведите ответ на задачу в формате одного числа на строку.

Система оценки

Группа	Баллы	Доп. ограничения	Необх. группы	Комментарий
		$n$		
0	1	–	–	Тесты из условия.
1	0,5	$n \leq 50$	0	
2	1,5	$n \leq 10^3$	0 – 1	
3	3	$n \leq 10^4$	0 – 2	
4	1	$n \leq 10^4$	0 – 3	Offline-проверка
5	1	$n \leq 11000$	0 – 4	Offline-проверка

Пример

Ввод

Вывод

7 3 2 1 9 5 4 6 8 0

1  
4  
6  
8

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