



Яндекс. Тренировки по алгоритмам июнь 2021, занятие 8

2 апр 2024, 05:11:25
старт: 23 июн 2021, 22:00:00
...

Объявления жюри

Положение участников Задачи Посылки

А. Высота дерева

	Все языки	Python 3.6
Ограничение времени	2 секунды	4 секунды
Ограничение памяти	64Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Реализуйте бинарное дерево поиска для целых чисел. Программа получает на вход последовательность целых чисел и строит из них дерево. Элементы в деревья добавляются в соответствии с результатом поиска их места. Если элемент уже существует в дереве, добавлять его не надо. Балансировка дерева не производится.

Формат ввода

На вход программа получает последовательность натуральных чисел. Последовательность завершается числом 0, которое означает конец ввода, и добавлять его в дерево не надо.

Формат вывода

Выведите единственное число – высоту получившегося дерева.

Пример

Ввод	Вывод
7 3 2 1 9 5 4 6 8 0	4

- ✓ А. Высота дерева
- ✓ В. Глубина добавляемых элементов
- ✓ С. Второй максимум
- ✓ D. Обход
- ✓ E. Вывод листьев
- ✓ F. Вывод развилок
- ✓ G. Вывод веток
- ✓ H. AVL-сбалансированность
- ✓ I. Родословная: число потомков
- ✓ J. Родословная: подсчет уровней

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь Отправить файл

```
1 from collections import deque
2
3 class BinarySearchTree():
4     def __init__(self, val=None, left=None, right=None) -> None:
5         if val is not None:
6             self.root = [val, left, right]
7         else:
8             self.root = None
9
10    def add(self, val, node=None, level=1):
11        if self.root is None:
12            self.root = [val, None, None]
13            return 1
14
15        if node is not None:
16            curr_node = node
17        else:
18            curr_node = self.root
19            level = 1
20
21        if val == curr_node[0]:
22            return 0
23        elif val < curr_node[0]:
24            if curr_node[1] is None:
25                curr_node[1] = [val, None, None]
26                return level+1
27            else:
28                return self.add(val, curr_node[1], level+1)
29        elif val > curr_node[0]:
30            if curr_node[2] is None:
31                curr_node[2] = [val, None, None]
32                return level+1
33            else:
34                return self.add(val, curr_node[2], level+1)
35
36    def getLength(self, node=None):
37        if node is None: return 0
38
```

Отправить *i* осталась 91 попытка

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы	
29 окт 2023, 00:09:02	95016584	A	Python 3.9 (PyPy 7.3.11)	OK	-	1.012s	41.28Mb	-	-	отчёт
28 окт 2023, 17:49:13	94968764	A	Python 3.9 (PyPy 7.3.11)	OK	-	1.08s	47.59Mb	-	-	отчёт
28 окт 2023, 17:37:19	94966614	A	Python 3.9 (PyPy 7.3.11)	OK	-	1.149s	47.49Mb	-	-	отчёт
27 окт 2023, 17:25:58	94828473	A	Python 3.9 (PyPy 7.3.11)	TL	-	2.077s	40.95Mb	31	-	отчёт
27 окт 2023, 17:12:43	94826524	A	Python 3.9 (PyPy 7.3.11)	TL	-	2.071s	42.80Mb	31	-	отчёт
27 окт 2023, 17:12:02	94826438	A	Python 3.12.1	TL	-	2.081s	4.26Mb	31	-	отчёт
27 окт 2023, 17:06:12	94825591	A	Python 3.12.1	TL	-	2.076s	4.25Mb	31	-	отчёт
27 окт 2023, 17:04:35	94825352	A	Python 3.9 (PyPy 7.3.11)	TL	-	2.073s	42.99Mb	31	-	отчёт
27 окт 2023, 16:59:53	94824673	A	Python 3.9 (PyPy 7.3.11)	WA	-	187ms	28.09Mb	2	-	отчёт