



Сдать решение задачи H

Полный балл:	4
Бонусные баллы:	
Ограничение времени:	1 с
Ограничение реального времени:	5 с
Ограничение памяти:	64М

Магическое двоичное дерево поиска

В этом задании требуется реализовать двоичное дерево поиска. Ваш класс `BinarySearchTree` должен поддерживать следующий интерфейс:

- Инициализатор, который опционально может принимать значение, которое будет находиться в корне дерева.
- `append(value)` добавляет в дерево новый элемент со значением `value`.
- Класс должен поддерживать конструкцию `value in tree`, где `value` — число, а `tree` — экземпляр класса дерева.
- Класс должен поддерживать механизм итерации. Итерирование соответствует обходу дерева в ширину.

Примеры

Входные данные

```
class BinarySearchTree:
    pass

if __name__ == '__main__':
    tree = BinarySearchTree()
    for v in [8, 3, 10, 1, 6, 4, 14, 13, 7]:
        tree.append(v)

    for v in [8, 12, 13]:
        print(v in tree)

    print(*tree)
```

Результат работы

```
True
False
True
8 3 10 1 6 14 4 7 13
```

Входные данные

```
class BinarySearchTree:
    pass

if __name__ == '__main__':
    tree = BinarySearchTree()
    for v in [5, 0, 6, 2, 1, 3]:
        tree.append(v)

    for v in [6, 12]:
        print(v in tree)

    print(*tree)
```

Результат работы

```
True
False
5 0 6 2 1 3
```

Примечания

В реализации обхода в ширину вам может пригодиться класс `deque` из модуля `collections`.

Сдать решение

Язык: python3 - Python3 3.6.5

Файл

Choose File

 No file chosen

Отправить!

Отправить!

Предыдущие решения этой задачи

Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть текст	исходный	Посмотреть протокол
1973	501:43:29	2443	Н	python3	OK	5	4	Просмотр		Просмотр
1972	501:14:40	2503	Н	python3	Неполное решение	4	0	Просмотр		Просмотр
1968	500:44:50	2496	Н	python3	Неполное решение	4	0	Просмотр		Просмотр
1966	500:38:24	2587	Н	python3	Неполное решение	3	0	Просмотр		Просмотр

- A
- B
- C
- D
- E
- F
- G
- H
- I