

-

« »  
• • • ( )

№1

« »  
:

. 9892 \_\_\_\_\_ ..  
\_\_\_\_\_ ..

-  
2022

# 1

```

    (map) - .
    unit- .
    :
```

1. insert( , ) —
2. remove( ) —
3. find( ) —
4. clear —
5. get\_keys —
6. get\_values —
7. print —

# 2

Python 3.8.  
: RBTreecolor, RBTreenode, RBTreecolor.

## 2.1 RBTreecolor

RBTreecolor .  
,

## 2.2 RBTreenode

RBTreenode .

python magic methods”( \_\_eq\_\_, \_\_lt\_\_, \_\_gt\_\_, etc. ),

:

- ””( ) property gparent.
- ””( , ) property uncle.
- ””( ) property bro.

color\_black() color\_red().  
: is\_black(), is\_red(), is\_left(),

is\_right().

## 2.3 RBTree

RBTree

"pythonic way"

"python magic methods".

- `__init__` — .
- `__getitem__` — .
- `__setitem__` — .
- `__delitem__` — .
- `__len__` — .
- `__str__` — .
- `__iter__` — ( ).
- `__bool__` — (False, , True)
- `height` —
- `get` — , .
- `items` —
- `keys` — .
- `values` — .
- `print_tree` — .
- `get_dot_string` — DOT ( ).
- `insert` — .
- `remove` — .
- `find` — .
- `clear` — .
- `get_keys` — .
- `get_values` — .

- `print` — .
- `_get_max_node` — .
- `_get_min_node` — .
- `_get_height` — .
- `_fix_insert` — .
- `_insert_case_1` — , 1.
- `_insert_case_2` — , 2.
- `_insert_case_3` — , 3.
- `_insert_case_4` — , 4.
- `_insert_case_5` — , 5.
- `_replace` — .
- `_delete` — (  $\leq 1$  )
- `_fix_delete` — .
- `_del_case_1` — , 1.
- `_del_case_2` — , 2.
- `_del_case_3` — , 3.
- `_del_case_4` — , 4.
- `_del_case_5` — , 5.
- `_del_case_6` — , 6.
- `_traverse_preorder` —
- `_traverse_inorder` — .
- `_traverse_postorder` — .
- `_left_rotate` — .
- `_right_rotate` — .
- `_print_tree` — .

- `_get_node` — .
- `_get_leaf` — - .
- `_swap_kv` — .

### 3

RBTree 1

1 – RBTree

<code>insert</code>	$O(\log n)$
<code>remove</code>	$O(\log n)$
<code>find</code>	$O(\log n)$
<code>clear</code>	$O(n)$
<code>get_keys</code>	$O(n)$
<code>get_values</code>	$O(n)$
<code>print</code>	$O(n)$

### 4 unit-

`check__tree(tree: RBTree, data: dict)`

, , .

`test__init(data)`

.

`test__insert(data, new__key, existing__key, new__value)`

.

`test__get(data)`

.

`test__get__error()`

,

.

`test__delete(data, existing__key)`

.

`test__delete__error(data, missing__key, error)`

`test__clear()`

## 5

```
1 examples = []
2
3
4 def example(name):
5     def decorator(f):
6         def wrapper(t):
7             print(f'Example {name}:')
8             print(f'Original tree: {t}')
9
10            f(t)
11
12            print('Tree:')
13            t.print_tree()
14            examples.append(wrapper)
15        return wrapper
16    return decorator
```

### 5.1

```
1 @example('find')
2 def example_find(t):
3     for k in t:
4         print(f'{k}: {t.find(k)}')
```

( . 1):

```

Example find:
Original tree: RBTree({0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9})
0: 0
1: 1
2: 2
3: 3
4: 4
5: 5
6: 6
7: 7
8: 8
9: 9
Tree:
--> 3:3 (BLACK)
    |
    |---L 1:1 (BLACK)
    |       |
    |       |---L 0:0 (BLACK)
    |       |---R 2:2 (BLACK)
    |---R 5:5 (BLACK)
    |       |
    |       |---L 4:4 (BLACK)
    |       |---R 7:7 (RED)
    |           |
    |           |---L 6:6 (BLACK)
    |           |---R 8:8 (BLACK)
    |               |
    |               |---R 9:9 (RED)

```

. 1 — ” ”

## 5.2

:

```

1 @example('insert')
2 def example_insert(t):
3     t.insert(1, 'new_value')
4     print('Replaced value at key 1 with "new_value":')
5     print(t)
6
7     t.insert('new_key', 'another_value')
8     print('Inserted value "new_value" with key "new_key":')
9     print(t)

```

( . 2):

```

Example insert:
Original tree: RBTree({0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9})
Replaced value at key 1 with "new_value":
RBTree({0: 0, 1: 'new_value', 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9})
Inserted value "new_value" with key "new_key":
RBTree({'new_key': 'another_value', 0: 0, 1: 'new_value', 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9})
Tree:
--> 3:3 (BLACK)
    |
    |---L 1:'new_value' (BLACK)
    |       |
    |       |---L 0:0 (BLACK)
    |       |---L 'new_key': 'another_value' (RED)
    |       |---R 2:2 (BLACK)
    |---R 5:5 (BLACK)
    |       |
    |       |---L 4:4 (BLACK)
    |       |---R 7:7 (RED)
    |           |
    |           |---L 6:6 (BLACK)
    |           |---R 8:8 (BLACK)
    |               |
    |               |---R 9:9 (RED)

```

. 2 — ” ”

## 5.3

:

```
1 @example('remove')
2 def example_remove(t):
3     print('Removed key 7:')
4     t.remove(7)
5     print(t)
```

( . 3):

```
Example remove:
Original tree: RBTre({'new_key': 'another_value', 0: 0, 1: 'new_value', 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9})
Removed key 7:
RBTre({'new_key': 'another_value', 0: 0, 1: 'new_value', 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 8: 8, 9: 9})
Tree:
--> 3:3 (BLACK)
    |
    |---L 1:'new_value' (BLACK)
    |   |
    |   |---L 0:0 (BLACK)
    |   |   |
    |   |   |---L 'new_key': 'another_value' (RED)
    |   |   |
    |   |   |---R 2:2 (BLACK)
    |   |
    |   |---R 5:5 (BLACK)
    |   |   |
    |   |   |---L 4:4 (BLACK)
    |   |   |   |
    |   |   |   |---R 8:8 (RED)
    |   |   |   |
    |   |   |   |---L 6:6 (BLACK)
    |   |   |   |   |
    |   |   |   |   |---R 9:9 (BLACK)
```

. 3 – ” ”

## 5.4

/

:

```
1 @example('keys and values')
2 def example_keys_and_values(t):
3     print('Keys:')
4     print(t.get_keys())
5
6     print('Values:')
7     print(t.get_values())
```

( . 4):

```
Example keys and values:
Original tree: RBTre({'new_key': 'another_value', 0: 0, 1: 'new_value', 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 8: 8, 9: 9})
Keys:
('new_key', 0, 1, 2, 3, 4, 5, 6, 8, 9)
Values:
('another_value', 0, 'new_value', 2, 3, 4, 5, 6, 8, 9)
Tree:
--> 3:3 (BLACK)
    |
    |---L 1:'new_value' (BLACK)
    |   |
    |   |---L 0:0 (BLACK)
    |   |   |
    |   |   |---L 'new_key': 'another_value' (RED)
    |   |   |
    |   |   |---R 2:2 (BLACK)
    |   |
    |   |---R 5:5 (BLACK)
    |   |   |
    |   |   |---L 4:4 (BLACK)
    |   |   |   |
    |   |   |   |---R 8:8 (RED)
    |   |   |   |
    |   |   |   |---L 6:6 (BLACK)
    |   |   |   |   |
    |   |   |   |   |---R 9:9 (BLACK)
```

. 4 – ” / ”

## 6

: <https://github.com/kira607/1lab-algo-3-2>