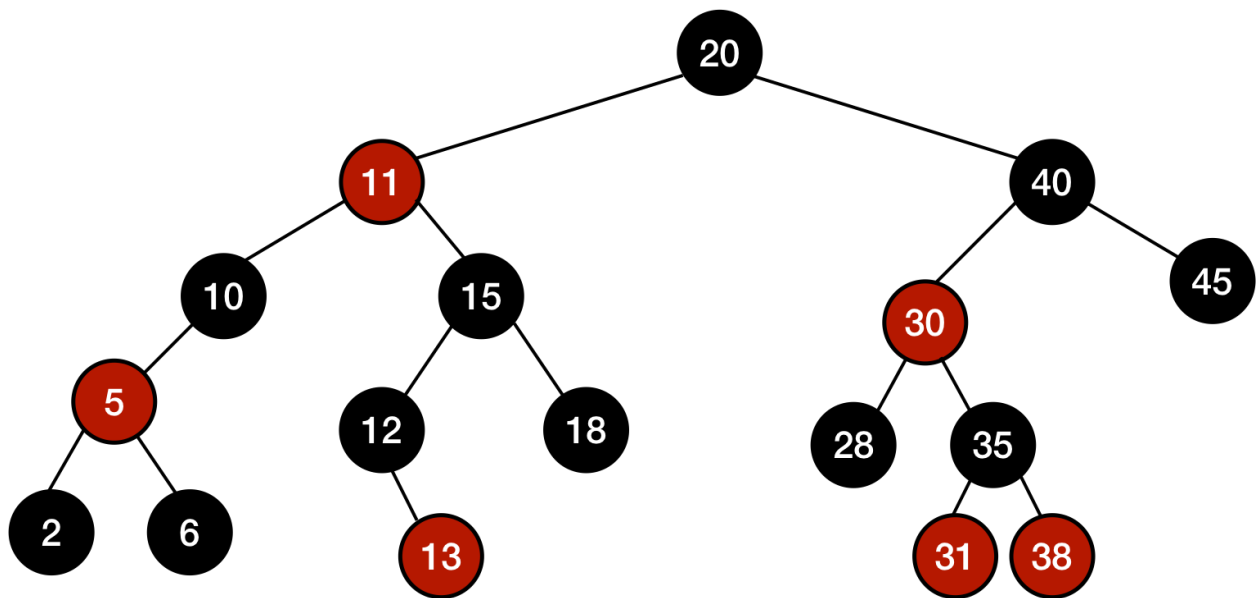


Red Black Tree Implementation

Implementing Red Black Tree & Treemap



Names :

Khadija Assem Saad (No. 27)

Nourhan Magdi Mohamed (No. 69)

Source Code :

<https://github.com/khadijaAssem/RedBlackTrees>

Content

- ❑ Introduction: page(3).
- ❑ IRedBlackTree interface: page(4).
- ❑ ITreeMap interface: page(5).
- ❑ Tests :page(7).

Introduction

Red Black Tree:

A redblack tree is a kind of self-balancing binary search tree in computer science. Each node of the binary tree has an extra bit, and that bit is often interpreted as the color (red or black) of the node. These color bits are used to ensure the tree remains approximately balanced during insertions and deletions. Balance is preserved by painting each node of the tree with one of two colors in a way that satisfies certain properties, which collectively constrain how unbalanced the tree can become in the worst case. When the tree is modified, the new tree is subsequently rearranged and repainted to restore the coloring properties. The properties are designed in such a way that this rearranging and recoloring can be performed efficiently

TreeMap:

A Red-Black tree based NavigableMap implementation. The map is sorted according to the natural ordering of its keys, or by a Comparator provided at map creation time, depending on which constructor is used. This implementation provides guaranteed $\log(n)$ time cost for the containsKey, get, put and remove operations. Algorithms are adaptations of those in Cormen, Leiserson, and Rivest's Introduction to Algorithms.

For the RedBlackTree interface implementation :

Functions:

| Function | Function | Running Time |
|-----------------------------------|--|--------------|
| INode<T,V> getRoot | Returns root of tree | O(1) |
| boolean isEmpty() | Returns true only if the tree is empty | O(1) |
| void clear() | Completely clears the tree | O(1) |
| V search(T key) | Returns the value associated with the given key | O(lg n) |
| boolean contains(T key) | Returns true only if the tree contains the given key | O(lg n) |
| void insert(T key,V value) | Inserts the given key and value (while keeping Red-Black Trees properties) or updates value if the key is present i the tree | O(lg n) |
| boolean delete(T key) | Deletes the element associated with the given key | O(lgn) |

For the TreeMap interface implementation:

Functions:

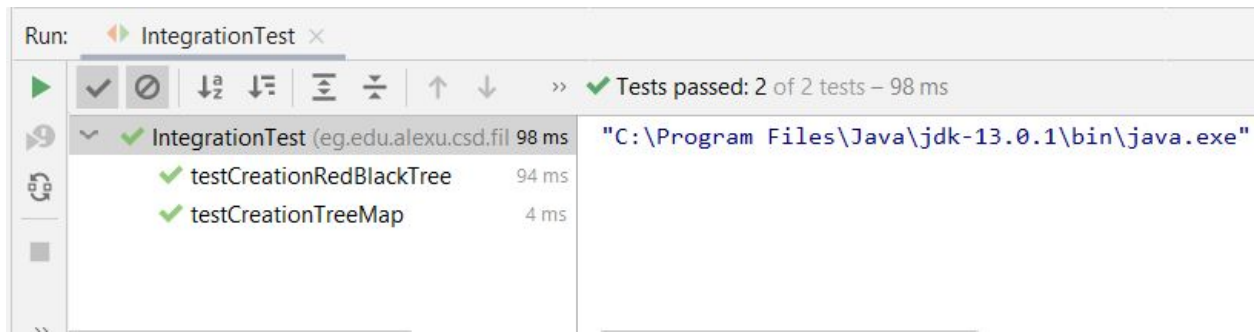
| Function | Function | Running Time |
|---|--|--------------|
| Map.Entry<T,V> ceilingEntry(T key) | Returns a key-value mapping associated with the least key greater than or equal to the given key, or null if there is no such key. | $O(\lg n)$ |
| T ceilingKey(T key) | Returns the least key greater than or equal to the given key, or null if there is no such key. | $O(\lg n)$ |
| clear() | Removes all of the mappings from this map. | $O(1)$ |
| containsKey(T key) | Returns true if this map contains a mapping for the specified key. | $O(\lg n)$ |
| containsValue(V value) | Returns true if this map maps one or more keys to the specified value. | $O(n)$ |
| Set<Map.Entry<T,V>> entrySet() | Returns a Set view of the mappings contained in this map in ascending key order. | $O(n)$ |
| Map.Entry<T,V> firstEntry() | Returns a key-value mapping associated with the least key in this map, or null if the map is empty. | $O(\lg n)$ |
| T firstKey() | Returns the first (lowest) key currently in this map, or null if the map is empty. | $O(\lg n)$ |
| Map.Entry<T,V> floorEntry(T key) | Returns a key-value mapping associated with the greatest key less than or equal to the given key, or null | $O(\lg n)$ |

| | | |
|---|--|------------|
| | if there is no such key. | |
| T floorKey(T key) | Returns the greatest key less than or equal to the given key, or null if there is no such key. | $O(\lg n)$ |
| V get(T key) | Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key. | $O(\lg n)$ |
| ArrayList<Map.Entry<T,V>> headMap(T toKey) | Returns a view of the portion of this map whose keys are strictly less than toKey in ascending order. | $O(n)$ |
| ArrayList<Map.Entry<T,V>> headMap(T toKey,boolean inclusive) | Returns a view of the portion of this map whose keys are less than (or equal to, if inclusive is true) toKey in ascending order. | $O(n)$ |
| Set<T> keySet() | Returns a Set view of the keys contained in this map. | $O(n)$ |
| Map.Entry<T,V> lastEntry() | Returns a key-value mapping associated with the greatest key in this map, or null if the map is empty. | $O(\lg n)$ |
| T lastKey() | Returns the last (highest) key currently in this map. | $O(\lg n)$ |
| Map.Entry<T,V> pollFirstEntry() | Removes and returns a key-value mapping associated with the least key in this map, or null if the map is empty. | $O(\lg n)$ |
| Map.Entry<T,V> pollLastEntry() | Removes and returns a key-value mapping associated with the greatest key in this map, or null if the map is empty. | $O(\lg n)$ |
| void put(T key,V value) | Associates the specified value with the specified key in this map. | $O(\lg n)$ |
| putAll (Map<T,V> map) | Copies all of the mappings from the specified map to this map. | $O(n)$ |
| boolean remove(T key) | Removes the mapping for this key from this TreeMap if present. return true if removed and false otherwise. | $O(\lg n)$ |

| | | |
|-------------------------------------|--|------------|
| int size() | Returns the number of key-value mappings in this map. | $O(\lg n)$ |
| Collection<V> values() | Returns a Collection view of the values contained in this map. | $O(n)$ |

Tests :

The two Integration tests are successfully passed



The 59 Unit Test are successfully passed

The screenshot shows an IDE's 'Run' window for a test suite named 'UnitTest'. The status bar at the top indicates 'Tests passed: 69 of 69 tests - 16 s 896 ms'. The test results list on the left shows 59 individual tests, all marked with a green checkmark and their execution times. The right pane shows the command used to run the tests: 'C:\Program Files\Java\jdk-13.0.1\bin\java.exe'. Below the command, there is a list of field names and their types, including 'field name interface eg.edu.alexu.csd.filestruct', 'field name interface java.util.Set', 'field name interface java.util.Set', and 'field name int'.

| Test Name | Execution Time |
|------------------------------------|----------------|
| testContainsAbsentKey | 93 ms |
| testGetElementInTreemap | 132 ms |
| testDeleteAbsentElementsInTreemap | 154 ms |
| testFloorEntryWithNull | 10 ms |
| testputAll | 6 ms |
| testClassContainsMapReference | 5 ms |
| testLastEntry | 9 ms |
| testStressContains | 6 s 333 ms |
| testContainsValueNormal | 40 ms |
| testpollFirstEntry | 6 ms |
| testEntrySet | 6 ms |
| testputWithNullValue | 9 ms |
| testRemoveNotFound | 6 ms |
| testUpdateValue | 2 ms |
| testSearchEmpty | 2 ms |
| testCeilingEntryWithNull | 2 ms |
| testpollLastEntry | 6 ms |
| testGetElementInTreemapNoDuplicate | 22 ms |
| testputAllWithNullValue | 2 ms |
| testInsertionWithNullKey | 2 ms |
| testIsEmptyTrue | 3 ms |
| testFirstEntry | 8 ms |
| testNormalInsertion | 3 ms |
| testClearTree | 3 ms |
| testContainsValueWithNullParameter | 6 ms |
| testGetRoot | 10 ms |
| testCeilingKey1 | 6 ms |
| testCeilingKey2 | 46 ms |
| testLastKey | 3 ms |
| testDeleteWhileInsertingInTreemap | 61 ms |
| testContainsKeyNotFound | 4 ms |

Run: UnitTest x

Tests passed: 69 of 69 tests - 16 s 896 ms

UnitTest (eg.edu.alexu.csd.fi 16 s 896 ms)

"C:\Program Files\Java\jdk-13.0.1\bin\java.exe"

field name interface eg.edu.alexu.csd.filestruct
field name interface java.util.Set
field name interface java.util.Set
field name int