## Difference between TreeSet and TreeMap in Java

The main difference between `TreeMap` and `TreeSet` is that `TreeMap` is an implementation of Map interface while `TreeSet` is an implementation of the `Set` interface. There are some similarities between both `TreeMap` and `TreeSet` and few differences as well. In this Java tutorial, we will first see similarities between `TreeMap` and `TreeSet`, and then you will learn some differences between `TreeMap` and `TreeSet` in Java.

The key point to remember about `TreeMap` and `TreeSet` is that they use `compareTo()` or `compare()` method to compare object, So if uses put a `String` object in `TreeSet` of Integers, `add()` method will throw `ClassCastException` at runtime prior to Java 5.

From Java 5 you can use Generics to avoid this happening by declaring `TreeMap` and `TreeSet` with parametrized version. If you want to master the Java Collection framework by heart, you can see the Java Generics and Collection book by Maurice Naftaline, one of the best works on the Java Collections framework.

## Similarities between TreeMap and TreeSet in Java

Here is a list of similarities between TreeMap and TreeSet in Java:

1. Both `TreeMap` and `TreeSet` are sorted data structures, which means they keep their element in predefined Sorted order. Sorting order can be natural sorting order defined by Comparable interface or custom sorting Order defined by Comparator interface.

Both `TreeMap` and `TreeSet` has overloaded constructor which accepts a Comparator, if provided all elements inside `TreeSet` or `TreeMap` will be compared and Sorted using this Comparator.

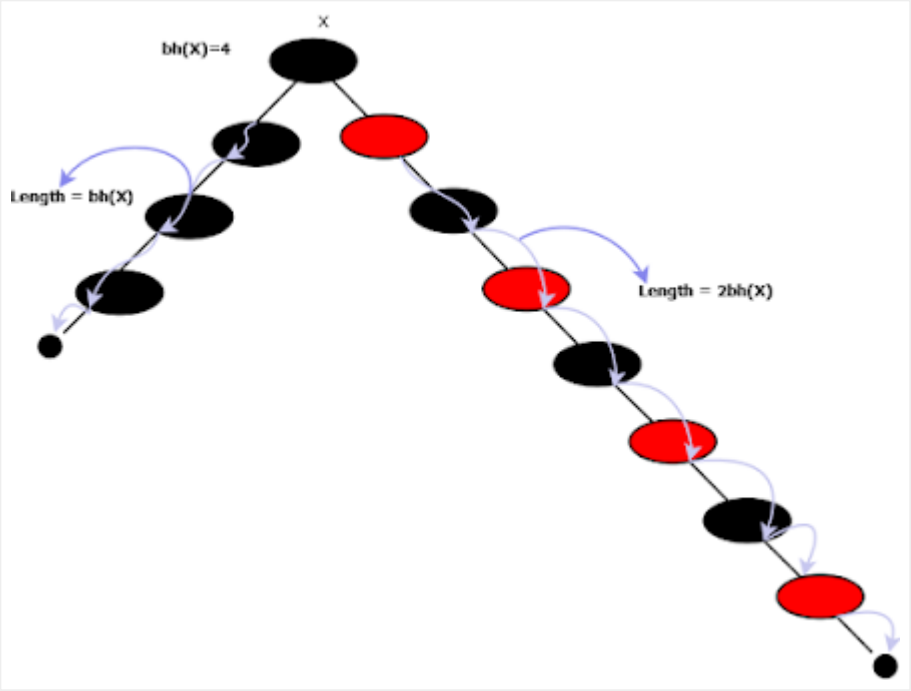
2. Both `TreeSet` and `TreeMap` implements base interfaces e.g. TreeSet implements Collection and Set interface so that they can be passed to a method where a Collection is expected and TreeMap implements java.util.Map interface, which means you can pass it when a Map is expected.

3. `TreeSet` is practically implemented using `TreeMap` instance, similar to `HashSet` which is internally backed by `HashMap` instance. See my post Internal Implementation of HashSet to learn more.

4. Both `TreeMap` and `TreeSet` are non-synchronized Collections, hence can not be shared between multiple threads. You can make both `TreeSet` and `TreeMap` synchronized by wrapping them into the Synchronized collection by calling `Collections.synchroinzedMap()` method.

5. Iterator returned by `TreeMap` and `TreeSet` are fail-fast, which means they will throw ConcurrentModificationException when `TreeMap` or `TreeSet` is modified structurally once Iterator is created. this fail-fast behavior is not guaranteed but works in the best effort.

6. Both `TreeMap` and `TreeSet` are slower than there Hash counterpart like `HashSet` and `HashMap` and instead of providing constant-time performance for add, remove, and get operation they provide performance in O(log(n)) order.




## TreeSet vs TreeMap in Java

Now let's see some differences between `TreeSet` vs `TreeMap` in Java:

1. Major difference between `TreeSet` and `TreeMap` is that `TreeSet` implements Set interface while `TreeMap` implements Map interface in Java.

2.Second difference between `TreeMap` and `TreeSet` is the way they store objects. `TreeSet` stores only one object while TreeMap uses two objects called key and Value. Objects in `TreeSet` are sorted while keys in `TreeMap` remain in sorted order.

3. Third difference between `TreeSet` and `TreeMap` is that, former implements `NavigableSet` while later implements `NavigableMap` in Java.

4. Fourth difference is that duplicate objects are not allowed in `TreeSet` but duplicates values are allowed in `TreeMap`.

That's all on the **difference between `TreeSet` and `TreeMap` in Java**. Both provide sorting but their usage is different. TreeMap is used to keep mappings between key and values in sorted order while TreeSet is used to keep just one element in sorted order. TreeSet also doesn't allow duplicates but TreeMap does allow duplicate values