

# Algorithms 2025/26: Tick 2

Your task is to implement the insert and delete algorithms for a Red-Black Tree of integers that is sorted using the ordinary '<' operator. We do not need to store payloads with the keys.

**Implement your algorithm** in Java. Your implementation should be contained in a single Java class which implements the interface below: the *insert(..)* and *delete(..)* methods are passed a Java object reference to the current root of the Red-Black tree. Both methods return a Java object reference to the new root, as it might be changed by an insert or deletion operation. Use `null` for the object reference to an empty tree, and represent leaf nodes in the tree with `null` references in their parent nodes. Your implementation will need to create new instances of the `RBNode` class and to set the `colour` field of new and existing `RBNodes` in the tree. The symbolic names `BLACK` and `RED` are provided for your convenience.

Your implementation should not require any libraries or external code, including the Java standard library (e.g. `java.util.*` and `java.io.*` classes).

Submit a single Java source file containing a class that implements the interface provided. Helper functions are allowed and encouraged to make your code readable! You may write unit tests and a class containing a `main(String[])` function in other files but are not asked to submit those.

**Submit >> [here](#) <<** You may re-submit if you wish. **The deadline is 12:00 on Thu 12 Mar 2026.**

```
interface Algs202526Tick2 {
    static final class RBNode {
        public static final boolean BLACK = false;
        public static final boolean RED = true;
        public boolean colour;
        public int key;
        public RBNode left, right;
    }
    RBNode insert(RBNode root, int key); // returns new root pointer
    RBNode delete(RBNode root, int key); // returns new root pointer
}
```

Save this ^^ into `Algs202526Tick2.java`, then write your solution in `Solution.java`:

```
public class Solution implements Algs202526Tick2 {...}
```