W3D4 Lab 3: BST Sort Objects

Description:

In this lab you'll use a BST to sort any type of object by providing a comparator

Provided:

We've provided the Code for the BinarySearchTree, RootNode, and Node classses.

Instructions:

1. Update the constructor for BinarySearchTree to take a comparator (and store it). Also update the constructor for Node and RootNode to take a comparator (and store it).

   Whenever a RootNode or Node is made it should receive the comparator for this tree from whomever is calling the constructor (The only one that makes RootNode is the BinarySearchTree itself, and Nodes are always made by the add() method in Node).

2. Update the add, contains, and remove methods inside the Node class to use the comparator function whenever they compare the values of the elements.
3. Test to see if your BinarySearchTree can now hold Employee objects by using the same Employee Data as before (shown below and on the next page).

   If you create a BST with the nameComparator it should see that Dave < George < Richard. If you create a BST with salaryComparator it should show that George < Richard < Dave. If you create a BST with a hireDateComparator it should show that George < Dave < Richard.

```
/**
 * Employee class
 */
class Employee {
   /**
    *
    * @param {String} name name of employee
    * @param {Number} salary monthly salary
    * @param {Number} year hired on year
    * @param {Number} month hired on month
    * @param {Number} day hired on day
    */
```

```
    constructor(name, salary, year, month, day) {
        this.name = name;
        this.salary = salary;
        this.hireDate = new Date(year, month - 1, day);
    }
}

let employees = [
    new Employee("George", 40000, 1996, 11, 5),
    new Employee("Dave", 50000, 2000, 1, 3),
    new Employee("Richard", 45000, 2001, 2, 7)
];
```