You will submit your solution to this assignment to the Curator System (as `HW1`). Your solution must be either a plain text file (e.g., NotePad) or a typed MS Word document; submissions in other formats will not be graded.

Partial credit will only be given if you show relevant work.

For questions 1 and 2, refer to the BST generic interface provided for Minor Project 2. Strive for the most efficient solution you can devise. You may not use any Java collections, including ones you implement yourself, to store any additional information (like copying the data elements from the BST into an array).

1. [25 points] Design an algorithm to determine whether a given binary tree is *full*, as defined in the course notes. Express your solution using Java code as if it were to be implemented as a member function the BST generic specified in Minor Project 2; your answer must conform to the public interface shown below:

```
// Pre:      none
// Post:     the BST is unchanged
// Returns:  true iff the BST is full
//
public boolean isFullBinaryTree( ) {

   // up to you. . .
}
```

It is perfectly acceptable to make use of a private helper function, in which case you must show implementations of both functions.

```
public boolean isFullBinaryTree() {

   return isFBTHelper(root);
}

private boolean isFBTHelper(BinaryNode sroot ) {

   // OK:  reached end of branch w/o a contradiction (only reach this
   //      if tree is empty):
   if ( sroot == null ) return true;

   // OK:  reached a leaf w/o a contradiction:
   if ( sroot.left == null && sroot.right == null ) return true;

   // NOPE:  reached a contradiction:
   if ( sroot.left == null || sroot.right == null ) return false;

   // ??:  at a 2-child node, must check subtrees:
   return isFBTHelper(sroot.left) && isFBTHelper(sroot.right);
}
```

2. [25 points] Design an algorithm to determine whether a BST contains any duplicate values; that is, answer the question: is there any value **X** such that **X** occurs in two different nodes within the tree. Assume that duplicates of an existing value will have been inserted to the right of the original copy.

Write an implementation of your algorithm, using Java syntax, as if the implementation were to be placed within the BST implementation from Minor Project 2. Your answer must conform to the public interface shown below:

```
// Pre:       none
// Post:      the BST is unchanged
// Returns:   true iff two different nodes in the BST contain values for
//            which equals() returns true
//
public boolean hasDuplicates( ) {

   // up to you. . .
}
```

It is perfectly acceptable to make use of a private helper function, in which case you must show implementations of both functions.

```
public boolean hasDuplicates() {

   return hasDupeHelper(root);
}

private boolean hasDupeHelper(BinaryNode sroot) {

   // nothing to see here, so no duplicate here:
   if ( sroot == null ) return false;

   // if value in current node is duplicated, then a duplicate MUST lie
   //    in the leftmost node of the right subtree; let's check that:
   BinaryNode rMin = findMin(sroot.right);
   if ( rMin != null && sroot.element.equals(rMin.element) ) return true;

   // if we get here, the current value is NOT duplicated; so we need to
   //    check both of the subtrees for duplicates (of other values):
   return hasDupeHelper(sroot.left) || hasDupeHelper(sroot.right);
}
```

3. A simple, unbucketed PR quadtree is used to organize data object that lie within a square bounded by the corners (0, 0) and (1024, 1024). Currently, the tree contains a single data object, which lies at the coordinates (25, 25). The following questions are independent.

a) [5 points] Give specific coordinates such that the insertion of a second data object at those coordinates will cause exactly one splitting operation to occur.

The object must lie at coordinates that are in a different quadrant of the world than the first one, which lies in the quadrant bounded by (0, 0) and (512, 512).

b) [10 points] Give specific coordinates such that the insertion of a second data object at those coordinates will cause exactly three splitting operation to occur.

The object must lie within the quadrant bounded by (0, 0) and (512, 512) to cause a single splitting. To cause a second splitting, it must lie within the SW quadrant of that quadrant, which is bounded by (0, 0) and (256, 256). To cause a third splitting, it must lie within the SW quadrant of that quadrant, which is bounded by (0, 0) and (128, 128). And, to NOT cause a fourth splitting it must not lie within the SW quadrant of that quadrant.

c) [10 points] Give specific coordinates such that the insertion of a second data object at those coordinates will cause exactly six splitting operation to occur.

Continuing in the same manner, a fourth splitting would occur if the object lies within the quadrant bounded by (0, 0) and (64, 64), and a fifth would occur if it lies within the quadrant bounded by (0, 0) and (32, 32). However, note that (25, 25) lies in the NE quadrant of that region, so to cause a sixth splitting, the object must lie within the quadrant bounded by (16, 16) and (32, 32); moreover, the original object lies within the NE quadrant of that region, so the new object must lie outside of that area.

4. [25 points] A simple, unbucketed PR quadtree is used to organize data object that lie within a square bounded by the corners (0, 0) and (1024, 1024). When the first two data objects are inserted, one splitting operation occurs. Neither of those data points lies on the boundary between two regions. What is the minimum distance the two data objects could be separated by? Why? Explain clearly.

Actually, you need to know whether the coordinates are integers or can be decimal values.

If the coordinates are integers, the two points could be a distance of 2 apart, directly along a line perpendicular to the boundary between two regions.

If the coordinates are decimal values, the two points could be arbitrarily close to a boundary without lying on it, and so in that case the best lower bound for their separation is 0, but there is no minimum.