You will submit your solution to this assignment to the Curator System (as `HW01`). 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.

If you work with a partner, be sure to put both the name and PID of each partner at the beginning of the file you submit.

Partial credit will only be given if you show relevant work and/or explanations.

---

1. [25 points] Design an algorithm to count and return the number of nodes in a binary tree that have two children. Express your solution as a pair of Java functions (not BST member functions), which would be implemented in the same package as the BST generic specified in Minor Project 2:

```
public int numFullNodes( BST<T> Tree ) {
   . . .
}
```

(Of course, the function shown above should have a recursive helper function.)

2. [25 points] Design an algorithm to count and return the least upper bound (LUB) of a data object `X`. Your solution will assume that the following public method has been added to the interface for the BST given in Minor Project 2:

```
// Pre:      X is a valid object of type T
//
// Returns:  reference to the unique object Y in the BST such that
//              Y = min { Z in tree | X.compareTo(Z) <= 0 }
//           or null if no such element exists in the BST
//
public T LUB(T X) {

   return LUBHelper(X, root, . . .);
}
```

Complete the implementation of the following private helper function, which would also be added to the given BST interface:

```
private T LUBHelper(T X, BinaryNode sroot, . . .) {
   . . .
}
```

The `". . ."` indicates you may use additional parameters if you find them useful or necessary. Your implementation should operate as efficiently as possible; that is, it should not examine any branch of the BST unless that branch could contain relevant data.

3. [25 points] Use Induction to prove the following fact: for every integer, $h \geq 0$, a full binary tree with h levels must have at least $2h-1$ nodes. (You may not use any of the tree theorems from the notes. An empty tree has 0 levels; a tree with only a root node has 1 level.)

4. Suppose we implement a BST generic in Java, that a Java reference requires 8 bytes of storage, and that we will use the BST to organize user data objects that each occupy 1 KiB of space (1024 bytes).

a) [10 points] Assume we implement the BST using a single node type (as in Project 2). If we store $2^{18}+1$ user data objects in the BST, and the resulting BST is a full binary tree, what is the ratio of space used for user data objects to the total space used for the BST?

b) [15 points] Repeat part a), but assume that we use two node types; internal nodes include two references for children but leaves do not include any references for children at all.