Assignment: BST Performance Test

A very straightforward assignment: test the performance of data retrieval in your own BST. No graphics this time.

The Task:

1.Create a BST project with classes Main, BST, Node.

Add the following methods :

- **public boolean insert(long value)** to insert a 'long' value into the BST. The method should return true if the value was inserted, false if not (= the value had already previously been inserted into the BST. BST values have to be unique (think Temple-ID)).
- **public int contains(long value)**. Returns -1 if the value is not in the BST, otherwise the number of search steps is returned (i.e. the level on which the target number resides in the tree).
- **public int getDepth().** Computes depth of tree.

It's up to you to implement these methods iteratively or recursively. (My personal choice, if that matters, would be iterative for the first two, recursive for the last.)

2. Test the performance of the tree in two steps:

a) insert the numbers from 1 ... 100000 *in sequential order.* Use the 'contains' method to check for every single of these numbers (100000 calls to 'contains(i)') and sum the resulting steps. Print the accumulated number of steps, and the depth of the tree. (before you program: what will this result be? you already know!)

b) insert the numbers 1 ... 100000 *in random order*. Again, check for each number and print the accumulated number of steps, as well as the depth of the tree. Before you start programming: How many steps do you expect?

Hint: the 1st test will use about 3000 times(!) more steps. Think why!