CSC373 HW3

Instructions:

1. Download the files TwoThreeIntSet.java and HW3Test.java from the submission folder for HW3.
2. Start up Eclipse with the same workspace and project as you used for HW1.
3. Inside the src folder, create a hw3 package
4. Select the files you downloaded and copy and paste them into the hw3 package in Eclipse
5. The JUnit library should still be there from HW1. If you did not complete HW1, you must add the JUnit library as described in the write-up for HW1.
6. You should now be able to edit the TwoThreeIntSet class. Finish implementing the functions contains and put. Each recursive function already handles the case when the node is a 2-node. You must add code (at the TODO comment) to handle the case when the node is a 3-node . Each TODO has some comments to help guide you. Make sure to read all the comments in the source file for extra instructions. In particular:
    - You are only allowed to change or add to the code in the recursive helper functions for put and contains. If you feel that you need to change code anywhere else, email your instructor with an explanation of the change you want to make and why you feel it is necessary/helpful. My response will most likely be no, but if I know WHY you want to make the change I might be able to provide some guidance.
    - You may not modify the function headers of any of the functions already present.
    - You may not add any fields to the TwoThreeIntSet class.
7. Like before, you will want to write additional tests in the HW3Test.java file. Make sure to take advantage of the levelOrder method that is already implemented in the TwoThreeIntSet class. It returns the level order of the tree as a String. This means you can easily write a number of tests by inserting into an initially empty tree, calling the levelOrder method, and making sure that the levelOrder returned is correct. This does require you to figure out what that level order should be by doing the insertions on paper first so that you can figure out what the tree should look like.
8. Once you are satisfied with your solution, submit the TwoThreeIntSet.java file in the appropriate submission folder. When you are prompted to drag a file into the submission folder, you should be able to drag it directly from inside Eclipse.
9. Double check your submission. Download it and make sure it is the file you intend to submit!

Hints:

When we discussed 2-3 trees, we talked about how you might temporarily introduce a 4-node when inserting into the tree, and that you fix this by moving the middle key up the tree. In my opinion, this is most easily handled by writing a recursive helper method that takes the root of a 2-3 tree and the item to insert and returns the root of the NEW tree with that item inserted. However, you should write your code in such a way that the tree returned might have a 4-node at the root. This means that when you build an answer, you can return a tree whose root is a 4-node, and when you get an answer back from a recursive call, you should check to see if you got a 4-node back and if you did, you can fix it at that point. The hints I provide in the comments assume that this is how you are solving the problem.

Note, that you do not have to keep the code I provided to you already in the put and contains methods if you would prefer to handle the 2-node case differently.

I suspect this is the most challenging of the assignments you have had so far. Make sure to give yourself plenty of time to develop, test, and debug. You can send me questions. If you send me code, it must be thoroughly commented with an explanation of how you are handling what I describe in the first paragraph of the Hints section.