Zillow Test

1d 23h
to test end

1

2

Search Tree Modification

Implement insert and delete in a modified-binary search tree of integers. This search tree is much like a binary search tree, but with three child nodes for each parent instead of two -- with the left node being values less than the parent, the right node values greater than the parent, and the middle nodes values equal to the parent. Delete only one node per delete method call.

Deleting the single remaining value in the tree should set the root node to null. Being asked to delete a value not in the tree should do nothing. Please do not throw exceptions.

For example, suppose I added the following nodes to the tree in this order: 5, 4, 9, 5, 7, 2, 2. The resulting tree would look like this:

```
     5
   / | \
  4  5  9
 /     /
2     7
|
2
```

If you would like to write your own custom test cases to run your code against, an example test case is:

insert 1,2,3
delete 2,1
insert 4

This test is equivalent to creating an empty tree, inserting 1, then inserting 2, then inserting 3, then deleting 2, then deleting 1, and finally inserting 4.
The output, for a correct tree will be 3,4 -- this representation is the sorted list of remaining values in the tree. **You do not have to write any code to print out this sorted list representation of a tree. The representation is merely to help with visualizing the contents of the tree (but not the structure).**

YOUR ANSWER

Zillow Test

1d 23h
to test end

1

2

Draft saved 02:07 pm

Original code

Python 2

```
class Node(object):  # Please do not remove or rename any of this code
    """Represents a single node in the Ternary Search Tree"""
    def __init__(self, val):
        self.val = val
        self.left = None
        self.mid = None
        self.right = None

class Tree(object):  # Please do not remove or rename any of this code
    """The Ternary Search Tree"""
    def __init__(self):
        self.root = None

    # Please complete this method.
    """Inserts val into the tree. There is no need to rebalance the tree."""
    def insert(self, val):
        pass

    # Please complete this method.
    """Deletes only one instance of val from the tree.
       If val does not exist in the tree, do nothing.
       There is no need to rebalance the tree."""
    def delete(self, val):
        pass

```

Line: 20 Col: 77

□ **Test against custom input**

Run Code

Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*