

btree

July 10, 2018

1 BTree Assignment

1.1 Overview

The purpose of this assignment is to write a pure Python implementation of a Btree and thus become familiar with the implementations details of the B-Tree. Below are the classes you will fill out.

```
In [23]: class BNode:
          # notes: http://book.pythontips.com/en/latest/\_\_slots\_\_magic.html
          __slots__ = ["tree", "contents", "children"]

          def __init__(self, tree, contents=None, children=None):
              pass

class BTree:
    def __init__(self, order):
        pass

    def insert(self, item):
        """Insert item into the tree. No return."""
        pass

    def remove(self, item):
        """Remove item from the tree. No return."""
        pass

    def __iter__(self):
        # notes: https://stackoverflow.com/questions/4019971/how-to-implement-iter-self
        yield BNode(self)

    def __contains__(self, item):
        """Returns true or false if item is in the tree"""

    @classmethod
    def bulkload(cls, items, order):
        """Performs a bulk load of items and returns a BTree object"""
        return BTree(order)
```

1.2 Here is the unit test that we'll use to verify your tree

```
In [24]: import unittest
import random

class BTreeTests(unittest.TestCase):
    def test_additions(self):
        bt = BTree(20)
        l = list(range(2000))
        for i, item in enumerate(l):
            bt.insert(item)
            self.assertEqual(list(bt), l[:i + 1])

    def test_bulkloads(self):
        bt = BTree.bulkload(list(range(2000)), 20)
        self.assertEqual(list(bt), list(range(2000)))

    def test_removals(self):
        bt = BTree(20)
        l = list(range(2000))
        list(map(bt.insert, l))
        rand = l[:]
        random.shuffle(rand)
        while l:
            self.assertEqual(list(bt), l)
            rem = rand.pop()
            l.remove(rem)
            bt.remove(rem)
        self.assertEqual(list(bt), l)

    def test_insert_regression(self):
        bt = BTree.bulkload(list(range(2000)), 50)

        for i in range(100000):
            bt.insert(random.randrange(2000))

In [27]: NOTEBOOK=True
if NOTEBOOK:
    unittest.main(argv=['first-arg-is-ignored'], exit=False)
else: # if you are running this from the command line in a .py file, then just run
    unittest.main()

FF.F
=====
FAIL: test_additions (__main__.BTreeTests)
-----
Traceback (most recent call last):
  File "<ipython-input-24-12127118887a>", line 10, in test_additions
    self.assertEqual(list(bt), l[:i + 1])
```

```
AssertionError: Lists differ: [<__main__._BNode object at 0x7fa0dce4f408>] != [0]
```

```
First differing element 0:
```

```
<__main__._BNode object at 0x7fa0dce4f408>
```

```
0
```

```
- [<__main__._BNode object at 0x7fa0dce4f408>]
```

```
+ [0]
```

```
=====
```

```
FAIL: test_bulkloads (__main__.BTreeTests)
```

```
-----
```

```
Traceback (most recent call last):
```

```
  File "<ipython-input-24-12127118887a>", line 14, in test_bulkloads
```

```
    self.assertEqual(list(bt), list(range(2000)))
```

```
AssertionError: Lists differ: [<__main__._BNode object at 0x7fa0dce4f148>] != [0, 1, 2, 3, 4, 5,
```

```
First differing element 0:
```

```
<__main__._BNode object at 0x7fa0dce4f148>
```

```
0
```

```
Second list contains 1999 additional elements.
```

```
First extra element 1:
```

```
1
```

```
Diff is 16937 characters long. Set self.maxDiff to None to see it.
```

```
=====
```

```
FAIL: test_removals (__main__.BTreeTests)
```

```
-----
```

```
Traceback (most recent call last):
```

```
  File "<ipython-input-24-12127118887a>", line 23, in test_removals
```

```
    self.assertEqual(list(bt), 1)
```

```
AssertionError: Lists differ: [<__main__._BNode object at 0x7fa0dce31c08>] != [0, 1, 2, 3, 4, 5,
```

```
First differing element 0:
```

```
<__main__._BNode object at 0x7fa0dce31c08>
```

```
0
```

```
Second list contains 1999 additional elements.
```

```
First extra element 1:
```

```
1
```

```
Diff is 16937 characters long. Set self.maxDiff to None to see it.
```

```
-----
```

```
Ran 4 tests in 0.644s
```

FAILED (failures=3)

1.3 Task

Your task is to implement a B-Tree using the above class and method stubs. Each day new progress should be made with a final complete solution turned in on Friday.