

Bug/progress report

He Zhang
zhangh7@oregonstate.edu

1. Description of Bugs I found

I have found four bugs in SUT. They are all in BTree class.

1.1 mid_order function

1.1.1 Function under test

The first one is in the mid_order function. This function is to output key values in the B-Tree into a list.

1.1.2 Tstl code for function

When I tested insert function in SUT I used mid_order function to build assertion in tstl as this:

```
~<BT>.insert(<int>) => \
    (len(<BT,1>.mid_order()) == pre<(len(<BT,1>.mid_order()))>+1)
```

1.1.3 Error when run tstl

When I ran tstl it reported error like this:

Traceback (most recent call last):

```
File "randomtester.py", line 469, in <module>
    main()
File "randomtester.py", line 330, in main
    if a[1]():
File                                     "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl-
master\generators\sut.py", line 15895, in guard632
    return (self.p_int[2]  !=  None)  and  (self.p_BT[1]  !=  None)  and
((len(self.p_BT[1].mid_order()) == 0) )
File                                     "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl-
master\generators\BTreenew.py", line 295, in mid_order
    if cur_node.is_leaf:
AttributeError: 'NoneType' object has no attribute 'is_leaf'
```

1.1.4 Error when run python

I tried in python, it also reported error:

Traceback (most recent call last):

```
File "BTreenew.py", line 621, in <module>
    print t.mid_order()
File "BTreenew.py", line 295, in mid_order
    if cur_node.is_leaf:
AttributeError: 'NoneType' object has no attribute 'is_leaf'
```

1.1.5 Bug and modify

I check the Source code and found min_order function does not consider the situation of empty B-Tree. So I added code below to modify this bug.

```
if (self.__size == 0):  
    return result
```

1.1.6 Check correctness of modification

After modified I re-ran tstl, it goes well, which means the bug is modified correctly. Here is the data after running `python randomtester.py -m 100`.

```
15.4330708661 PERCENT COVERED  
9.62599992752 TOTAL RUNTIME  
100 EXECUTED  
10000 TOTAL TEST OPERATIONS  
6.71798920631 TIME SPENT EXECUTING TEST OPERATIONS  
2.65600919724 TIME SPENT EVALUATING GUARDS AND CHOOSING  
ACTIONS  
0.00699973106384 TIME SPENT CHECKING PROPERTIES  
6.72498893738 TOTAL TIME SPENT RUNNING SUT  
0.102000713348 TIME SPENT RESTARTING  
0.0 TIME SPENT REDUCING TEST CASES  
91 BRANCHES COVERED  
67 STATEMENTS COVERED
```

1.2 Other bugs

Other three bugs are alike bug in `mid_order` function, which is caused by lack of considering the situation of empty B-Tree. They are located in `delete()`, `search()`, `min()` and `max()` functions. I attach a form which records and analyzes these bugs in the end of the report.

2. Progress to Date

I have made a great progress in writing `tstl` file and testing SUT.

I have tested `mid_order()`, `insert()`, `delete()`, `search()`, `min()` and `max()` functions directly. And during test these functions, they also call `__init__()`, `tree_size()`, `__split()`, `full()`, `search()`, `__search()`, `__delete()`, `__successor()`, `__check_brother_borrow()`, `__merge_brother()` functions, so these functions are also tested indirectly.

I have tested `Node` and `BTree` classes.

Now the coverage of test is 22.677%, which in the beginning the coverage of test is about 15%.

3. Estimate Progress by End of Term

My expectation for this project is to find more bugs and get a higher coverage. So by the end of term I hope I can find another 3-4 bugs and get a coverage higher than 30%.

4. Discuss Quality of SUT

The quality of this software under test is not so good. Till now I have found 4 bugs. It has about 580 lines. So the bugs per thousand lines is about 6.89. This is not a low rate. Also, considering there are still some lines I have not tested yet, the rate may go higher in the future.

The main problem of this SUT is that it always forget to consider the special cases, for example empty B-Tree. This is a normal problem for software. Except this, the main feature of SUT is just good. The insert, delete and other operation works correctly.

5. Code Coverage

As shown above, the coverage of test until now is 22.677%. This is a fair number for this SUT, since it has a lot of branches, functions and classes.

CS 562 Project Part 3

Form of Bugs

function	Tstl code	Tstl running error	Bug and Modification	Regression Test
delete() search()	(len(<BT,l>.mid_order()) == 0) -> ~<BT>.delete(<int>) => \ (len(<BT,l>.mid_order()) == pre<(len(<BT,l>.mid_order()))>)	ERROR: (<type 'exceptions.TypeError'>, TypeError("object of type 'NoneType' has no len()"), <traceback object at 0x04 7D2F58>) TRACEBACK: File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\sut.py", line 21569, in safely act[2]() File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\sut.py", line 15716, in act626 self.p_BT[0].delete(self.p_int[2]) File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\BTreenew.py", line 342, in delete node, index = self.search(key) File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\BTreenew.py", line 248, in search return self.__search(self.__root, instance) File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\BTreenew.py", line 260, in __search cur_len = len(cur_node)	Forget to consider special case. In delete() function, adding: if (self.__size == 0): return False in search() function, adding: if (self.__size == 0): return None, None	20.9702660407 PERCENT COVERED 9.03599977493 TOTAL RUNTIME 100 EXECUTED 10000 TOTAL TEST OPERATIONS 6.16499853134 TIME SPENT EXECUTING TEST OPERATIONS 2.53400087357 TIME SPENT EVALUATING GUARDS AND CHOOSING ACTIONS 0.00200009346008 TIME SPENT CHECKING PROPERTIES 6.1669986248 TOTAL TIME SPENT RUNNING SUT 0.112000226974 TIME SPENT RESTARTING 0.0 TIME SPENT REDUCING TEST CASES 125 BRANCHES COVERED 93 STATEMENTS COVERED
min()	(len(<BT,l>.mid_order()) == 0) -> ~<BT>.min() => \ ((<BT,l>.min()) == None)	ERROR: (<type 'exceptions.AttributeError'>, AttributeError("'NoneType' object has no attribute 'is_leaf'",), <tracebac k object at 0x0496CEB8>) TRACEBACK: File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\sut.py", line 21908, in safely act[2]() File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\sut.py", line 16091, in act640 self.p_BT[1].min() File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl-	Forget to consider special case. In min() function, adding: if self.__root is None: return None	21.2832550861 PERCENT COVERED 1.01899981499 TOTAL RUNTIME 10 EXECUTED 1000 TOTAL TEST OPERATIONS 0.566999673843 TIME SPENT EXECUTING TEST OPERATIONS 0.286999940872 TIME SPENT EVALUATING GUARDS AND CHOOSING ACTIONS 0.0 TIME SPENT CHECKING PROPERTIES 0.566999673843 TOTAL TIME SPENT RUNNING SUT

CS 562 Project Part 3

		master\generators\BTreenew.py", line 277, in min while not cur_node.is_leaf:		0.063000202179 TIME SPENT RESTARTING 0.0 TIME SPENT REDUCING TEST CASES 127 BRANCHES COVERED 94 STATEMENTS COVERED
max()	(len(<BT,1>.mid_order()) == 0) -> ~<BT>.max() => \ ((<BT,1>.max()) == None)	ERROR: (<type 'exceptions.AttributeError'>, AttributeError("'NoneType' object has no attribute 'is_leaf'",), <tracebac k object at 0x042523F0>) TRACEBACK: File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\sut.py", line 22118, in safely act[2]() File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\sut.py", line 16216, in act645 self.p_BT[0].max() File "E:\Courses\2015-2016\Winter\CS562\tstl-master\tstl- master\generators\BTreenew.py", line 285, in max while not cur_node.is_leaf:	Forget to consider special case. In max() function, adding: if self.__root is None: return None	22.5352112676 PERCENT COVERED 9.25 TOTAL RUNTIME 100 EXECUTED 10000 TOTAL TEST OPERATIONS 6.3119893074 TIME SPENT EXECUTING TEST OPERATIONS 2.66301035881 TIME SPENT EVALUATING GUARDS AND CHOOSING ACTIONS 0.0260007381439 TIME SPENT CHECKING PROPERTIES 6.33799004555 TOTAL TIME SPENT RUNNING SUT 0.0819997787476 TIME SPENT RESTARTING 0.0 TIME SPENT REDUCING TEST CASES 134 BRANCHES COVERED 99 STATEMENTS COVERED