COP3530 17F Project 2

Grayson Arneson

15152354

Section #23HC

By submitting this document, I affirm that all the work submitted is my solely my own and that in completing this project I did nothing contrary to either the the spirit or the letter of the UF Honor Code

Section 1

Implementation Summary

	BST_LEAF	BST_ROOT	BST_RAND	AVL	HASH_OPEN	HASH_BUCKET
My template class has the required name, file extension, and template parameters. **[y/n]**	y	y	y	y	n	n
My template class was implemented *efficiently* using the required technique as described in the project description. **[y/n]**	y	y	y	y	n	n
I have implemented and throughly tested each of the **required** methods and they ***all*** behave correctly. **[y/n]**	y	y	y	y	n	n
AVL-based map only: I have throughly tested and verified that at the completion of an insertion/removal that the tree and all of its subtrees are AVL balanced.	—	—	—	y	—	—
I have implemented and throughly tested each of the **Group 1** methods and they ***all*** behave correctly (bonus). **[y/n]**	n	n	n	n	n	n
I have implemented and throughly tested *efficient* iterators (both **const** and non-const) over a map instance's data and my map supports the bonus **Group 2** methods iterator creation operations (bonus). **[y/n]**	n	n	n	n	n	n
I have verified by testing that a **const** instance of my map class and the data it holds cannot be modified, either through the map operations nor via iterator over the map's elements. **[y/n]**	n	n	n	n	n	n
I wrote my tests using CATCH. **[y/n]**	y	y	y	y	n	n
I have verified my map class is memory-leak free using valgrind. **[y/n]**	y	y	y	y	n	n
I certify that all of the responses I have given for this list class are TRUE [*your initials*]	GA	GA	GA	GA	GA	GA

Section 2

The easiest part of the assignment was the lookup function, since it essentially just checked if we understood the basic theory behind a BST. The most difficult part was the remove. Though it was almost universal for the various BSTs, finding all the edge cases for removal (leaf, one child, two children, etc.) took an extremely long amount of time. I believe that the assignment was meant to teach us the fundamentals of BSTs, how to implement them efficiently, and how to use them to their full potential. I believe the project succeeded in its goal of doing that. I don't have any improvements to suggest.

Section 3

Note: Valgrind output is collective for all subsections

BSTLEAF

```
g++ --std=c++11 Project2.cpp && ./a.out
===============================================================
===============
All tests passed (12 assertions in 4 test cases)
==10992==
==10992== HEAP SUMMARY:
==10992==      in use at exit: 72,704 bytes in 1 blocks
==10992==   total heap usage: 349 allocs, 348 frees, 109,140
bytes allocated
==10992==
==10992== LEAK SUMMARY:
==10992== definitely lost: 0 bytes in 0 blocks
==10992== indirectly lost: 0 bytes in 0 blocks
==10992==      possibly lost: 0 bytes in 0 blocks
==10992== still reachable: 72,704 bytes in 1 blocks
==10992==      suppressed: 0 bytes in 0 blocks
==10992== Rerun with --leak-check=full to see details of leaked
memory
==10992==
==10992== For counts of detected and suppressed errors, rerun
with: -v
==10992== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0
from 0)
lin309-10:16%
```

BSTROOT

```
g++ --std=c++11 Project2.cpp && ./a.out
===============================================================
===============

All tests passed (12 assertions in 4 test cases)

==10992==
==10992== HEAP SUMMARY:
==10992==      in use at exit: 72,704 bytes in 1 blocks
==10992==   total heap usage: 349 allocs, 348 frees, 109,140
bytes allocated
==10992==
==10992== LEAK SUMMARY:
==10992== definitely lost: 0 bytes in 0 blocks
==10992== indirectly lost: 0 bytes in 0 blocks
```

```
==10992==      possibly lost: 0 bytes in 0 blocks
==10992== still reachable: 72,704 bytes in 1 blocks
==10992==      suppressed: 0 bytes in 0 blocks
==10992== Rerun with --leak-check=full to see details of leaked
memory
==10992==
==10992== For counts of detected and suppressed errors, rerun
with: -v
==10992== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0
from 0)
lin309-10:16%
```

BSTRAND

```
g++ --std=c++11 Project2.cpp && ./a.out

===============================================================
==============

All tests passed (12 assertions in 4 test cases)


==10992==
==10992== HEAP SUMMARY:
==10992==      in use at exit: 72,704 bytes in 1 blocks
==10992==   total heap usage: 349 allocs, 348 frees, 109,140
bytes allocated
==10992==
==10992== LEAK SUMMARY:
==10992== definitely lost: 0 bytes in 0 blocks
==10992== indirectly lost: 0 bytes in 0 blocks
==10992==      possibly lost: 0 bytes in 0 blocks
==10992== still reachable: 72,704 bytes in 1 blocks
==10992==      suppressed: 0 bytes in 0 blocks
==10992== Rerun with --leak-check=full to see details of leaked
memory
==10992==
==10992== For counts of detected and suppressed errors, rerun
with: -v
==10992== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0
from 0)
lin309-10:16%
```

AVL

```
g++ --std=c++11 Project2.cpp && ./a.out
```

```
===============================================================
==============

All tests passed (12 assertions in 4 test cases)


==10992==
==10992== HEAP SUMMARY:
==10992==      in use at exit: 72,704 bytes in 1 blocks
==10992==   total heap usage: 349 allocs, 348 frees, 109,140
bytes allocated
==10992==
==10992== LEAK SUMMARY:
==10992== definitely lost: 0 bytes in 0 blocks
==10992== indirectly lost: 0 bytes in 0 blocks
==10992==      possibly lost: 0 bytes in 0 blocks
==10992== still reachable: 72,704 bytes in 1 blocks
==10992==      suppressed: 0 bytes in 0 blocks
==10992== Rerun with --leak-check=full to see details of leaked
memory
==10992==
==10992== For counts of detected and suppressed errors, rerun
with: -v
==10992== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0
from 0)
lin309-10:16%
```