Name: ***Jonathan Smoley***

Assignment: 2

Complete & Correct:	39 / 42
Tests:	6 / 6
Format and Comments:	6 / 6
Writeup:	6 / 6
Total Score:	57 / 60

Comments:

1. (-3) op[] needs special case for accessing last element in list (tail case)
2. In copy assignment, calling rhs[i] is inefficient. You will need to fix this for HW-4
3. Note you don't need to use `this->` except for in very specific situations (nothing in this assignment)

Unit test output:

```
==159731== Memcheck, a memory error detector
==159731== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==159731== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==159731== Command: ./hw2_test
==159731==
[==========] Running 20 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 20 tests from BasicLinkedSeqTests
[ RUN      ] BasicLinkedSeqTests.EmptySeqSize
[       OK ] BasicLinkedSeqTests.EmptySeqSize (11 ms)
[ RUN      ] BasicLinkedSeqTests.EmptySeqContains
[       OK ] BasicLinkedSeqTests.EmptySeqContains (1 ms)
[ RUN      ] BasicLinkedSeqTests.EmptySeqMemberAccess
[       OK ] BasicLinkedSeqTests.EmptySeqMemberAccess (28 ms)
[ RUN      ] BasicLinkedSeqTests.AddAndCheckSize
[       OK ] BasicLinkedSeqTests.AddAndCheckSize (6 ms)
[ RUN      ] BasicLinkedSeqTests.AddAndCheckContains
[       OK ] BasicLinkedSeqTests.AddAndCheckContains (5 ms)
[ RUN      ] BasicLinkedSeqTests.OutOfBoundsInsertIndexes
[       OK ] BasicLinkedSeqTests.OutOfBoundsInsertIndexes (3 ms)
[ RUN      ] BasicLinkedSeqTests.EraseAndCheckSize
[       OK ] BasicLinkedSeqTests.EraseAndCheckSize (10 ms)
[ RUN      ] BasicLinkedSeqTests.EraseAndCheckContains
[       OK ] BasicLinkedSeqTests.EraseAndCheckContains (7 ms)
[ RUN      ] BasicLinkedSeqTests.OutOfBoundsEraseIndexes
[       OK ] BasicLinkedSeqTests.OutOfBoundsEraseIndexes (4 ms)
[ RUN      ] BasicLinkedSeqTests.CheckClear
[       OK ] BasicLinkedSeqTests.CheckClear (1 ms)
[ RUN      ] BasicLinkedSeqTests.DestructorNoThrowChecksWithNew
[       OK ] BasicLinkedSeqTests.DestructorNoThrowChecksWithNew (2 ms)
[ RUN      ] BasicLinkedSeqTests.CopyConstructorChecks
[       OK ] BasicLinkedSeqTests.CopyConstructorChecks (9 ms)
[ RUN      ] BasicLinkedSeqTests.MoveConstructorChecks
[       OK ] BasicLinkedSeqTests.MoveConstructorChecks (7 ms)
```

```
[ RUN      ] BasicLinkedSeqTests.CopyAssignmentOpChecks
[       OK ] BasicLinkedSeqTests.CopyAssignmentOpChecks (9 ms)
[ RUN      ] BasicLinkedSeqTests.CheckRValueAccess
[       OK ] BasicLinkedSeqTests.CheckRValueAccess (2 ms)
[ RUN      ] BasicLinkedSeqTests.CheckLValueAccess
[       OK ] BasicLinkedSeqTests.CheckLValueAccess (2 ms)
[ RUN      ] BasicLinkedSeqTests.CheckConstRValueAccess
[       OK ] BasicLinkedSeqTests.CheckConstRValueAccess (2 ms)
[ RUN      ] BasicLinkedSeqTests.OutOfBoundsLValueAccess
[       OK ] BasicLinkedSeqTests.OutOfBoundsLValueAccess (2 ms)
[ RUN      ] BasicLinkedSeqTests.OutOfBoundsRValueAccess
[       OK ] BasicLinkedSeqTests.OutOfBoundsRValueAccess (2 ms)
[ RUN      ] BasicLinkedSeqTests.StringInsertionChecks
[       OK ] BasicLinkedSeqTests.StringInsertionChecks (9 ms)
[----------] 20 tests from BasicLinkedSeqTests (142 ms total)

[----------] Global test environment tear-down
[==========] 20 tests from 1 test suite ran. (182 ms total)
[  PASSED  ] 20 tests.
==159731==
==159731== HEAP SUMMARY:
==159731==     in use at exit: 0 bytes in 0 blocks
==159731==   total heap usage: 683 allocs, 683 frees, 150,519 bytes allocated
==159731==
==159731== All heap blocks were freed -- no leaks are possible
==159731==
==159731== For lists of detected and suppressed errors, rerun with: -s
==159731== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Performance test output:

```
# All times in milliseconds (msec)
# Column 1 = input data size
# Column 2 = avg time linked-list insert front
# Column 3 = avg time cpp-list insert front
# Column 4 = avg time linked-list insert middle
# Column 5 = avg time cpp-list insert middle
# Column 6 = avg time linked-list insert end
# Column 7 = avg time cpp-list insert end
# Column 8 = avg time linked-list update front
# Column 9 = avg time cpp-list update front
# Column 10 = avg time linked-list update middle
# Column 11 = avg time cpp-list update middle
# Column 12 = avg time linked-list update end
# Column 13 = avg time cpp-list update end
# Column 14 = avg time linked-list erase front
# Column 15 = avg time cpp-list erase front
# Column 16 = avg time linked-list erase middle
# Column 17 = avg time cpp-list erase middle
# Column 18 = avg time linked-list erase end
# Column 19 = avg time cpp-list erase end
0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
10000 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
20000 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
30000 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
40000 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
50000 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
60000 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
70000 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
80000 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.56 0.00 0.00 0.00 0.00 0.00 0.11 0.00
90000 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.56 0.00 0.00 0.00 0.00 0.00 1.00 0.00
```

100000	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.11	0.00	0.00	0.00	0.00	0.00	1.00	0.00
110000	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
120000	0.00	0.00	0.33	0.22	0.00	0.00	0.00	0.00	0.22	1.22	1.22	0.00	0.00	0.00	0.00	0.22	1.33	0.00
130000	0.00	0.00	0.33	0.44	0.00	0.00	0.00	0.00	0.22	1.67	1.56	0.00	0.00	0.00	0.33	0.22	1.44	0.00
140000	0.00	0.00	0.33	0.89	0.00	0.00	0.00	0.00	0.44	1.89	2.11	0.00	0.00	0.00	0.22	0.67	2.00	0.00
150000	0.00	0.00	0.56	1.00	0.00	0.00	0.00	0.00	0.78	2.00	2.22	0.00	0.00	0.00	0.89	1.11	2.11	0.00
160000	0.00	0.00	1.11	1.22	0.00	0.00	0.00	0.00	1.00	2.11	3.22	0.00	0.00	0.00	1.00	1.11	3.11	0.00
170000	0.00	0.00	1.11	1.11	0.00	0.00	0.00	0.00	1.11	2.78	3.89	0.00	0.00	0.00	1.11	1.33	4.22	0.00
180000	0.00	0.00	1.11	1.22	0.00	0.00	0.00	0.00	1.22	3.11	4.22	0.00	0.00	0.00	1.00	1.33	4.11	0.00
190000	0.00	0.00	1.33	1.56	0.00	0.00	0.00	0.00	1.22	3.67	5.78	0.00	0.00	0.00	1.33	1.44	5.33	0.00
200000	0.00	0.00	2.11	2.78	0.00	0.00	0.00	0.00	2.56	4.67	6.89	0.00	0.00	0.00	2.11	2.33	6.22	0.00