CS3500: Object-Oriented Design Spring 2014

Class 11

2.14.2014

Today...

- Readings
- Assignments
- Midterm Review

Readings

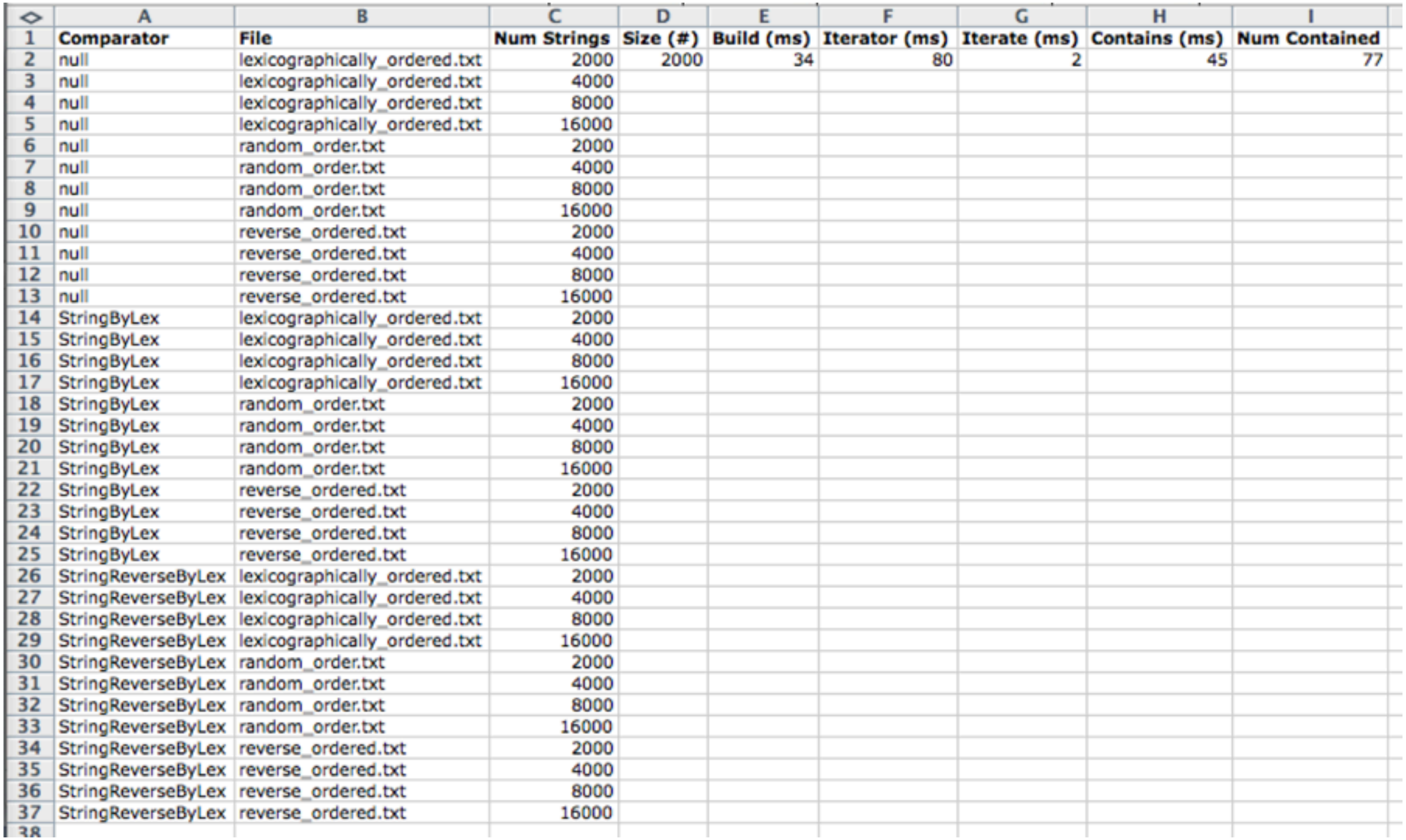
Please read the following paper for class on Friday, February 21, 2014:

> Chris Okasaki, "Red-black trees in a functional setting," Journal of Functional Programming, 9(4), pages 474-477, July 1999.

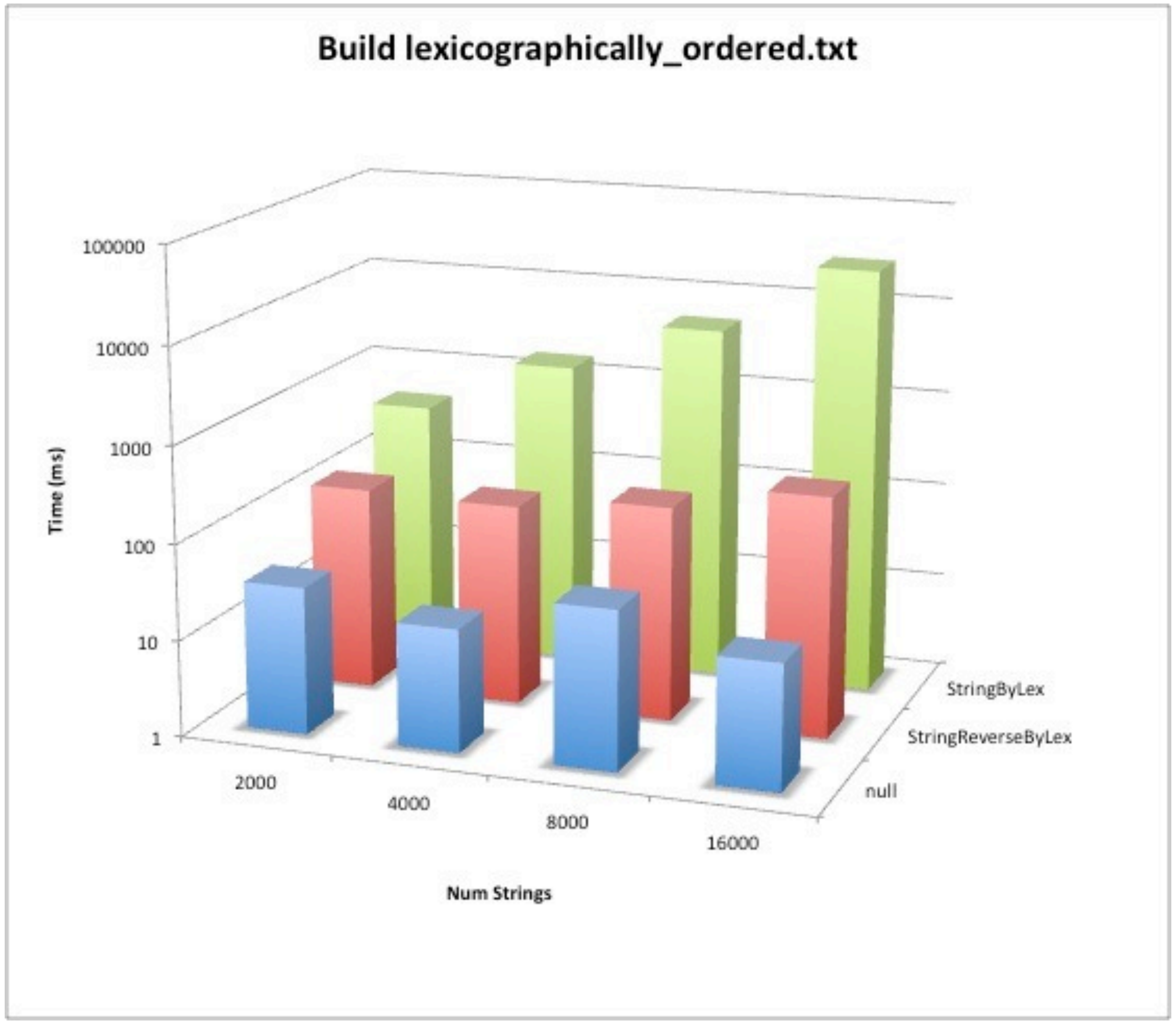
Link to paper on course website.

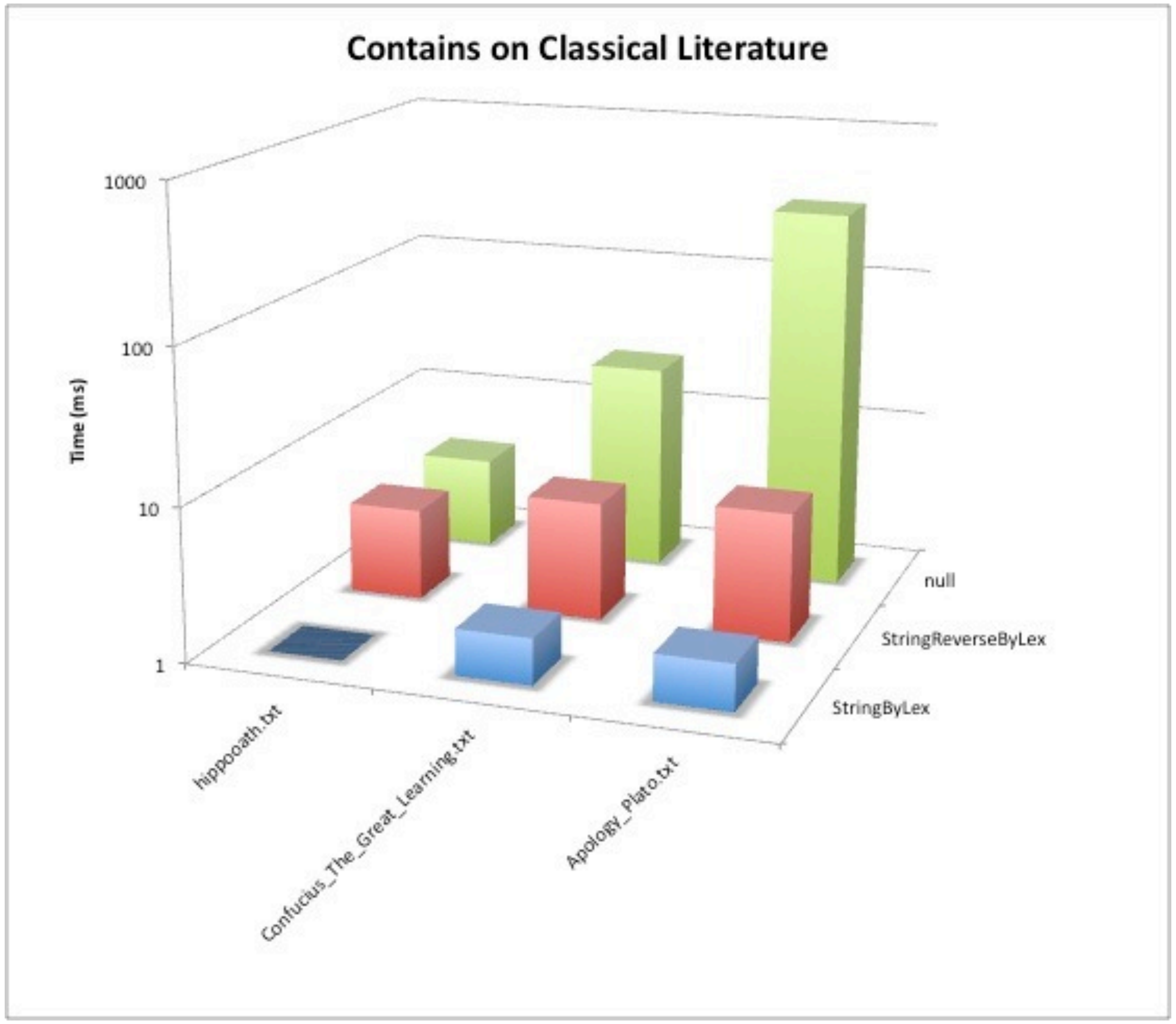
Assignment 6

- Timing program for MyMap<K,V>
- Due: Friday, February 14, 2014 at 11:59pm



	A	B	C	D	E	F	G	H	I
1	Comparator	File	Num Strings	Size (#)	Build (ms)	Iterator (ms)	Iterate (ms)	Contains (ms)	Num Contained
2	null	lexicographically_ordered.txt	2000	2000	34	80	2	45	77
3	null	lexicographically_ordered.txt	4000						
4	null	lexicographically_ordered.txt	8000						
5	null	lexicographically_ordered.txt	16000						
6	null	random_order.txt	2000						
7	null	random_order.txt	4000						
8	null	random_order.txt	8000						
9	null	random_order.txt	16000						
10	null	reverse_ordered.txt	2000						
11	null	reverse_ordered.txt	4000						
12	null	reverse_ordered.txt	8000						
13	null	reverse_ordered.txt	16000						
14	StringByLex	lexicographically_ordered.txt	2000						
15	StringByLex	lexicographically_ordered.txt	4000						
16	StringByLex	lexicographically_ordered.txt	8000						
17	StringByLex	lexicographically_ordered.txt	16000						
18	StringByLex	random_order.txt	2000						
19	StringByLex	random_order.txt	4000						
20	StringByLex	random_order.txt	8000						
21	StringByLex	random_order.txt	16000						
22	StringByLex	reverse_ordered.txt	2000						
23	StringByLex	reverse_ordered.txt	4000						
24	StringByLex	reverse_ordered.txt	8000						
25	StringByLex	reverse_ordered.txt	16000						
26	StringReverseByLex	lexicographically_ordered.txt	2000						
27	StringReverseByLex	lexicographically_ordered.txt	4000						
28	StringReverseByLex	lexicographically_ordered.txt	8000						
29	StringReverseByLex	lexicographically_ordered.txt	16000						
30	StringReverseByLex	random_order.txt	2000						
31	StringReverseByLex	random_order.txt	4000						
32	StringReverseByLex	random_order.txt	8000						
33	StringReverseByLex	random_order.txt	16000						
34	StringReverseByLex	reverse_ordered.txt	2000						
35	StringReverseByLex	reverse_ordered.txt	4000						
36	StringReverseByLex	reverse_ordered.txt	8000						
37	StringReverseByLex	reverse_ordered.txt	16000						
38									


Build lexicographically_ordered.txt
Time (ms)
100000
10000
1000
100
10
1
2000
4000
8000
16000
Num Strings
StringByLex
StringReverseByLex
null



Contains on Classical Literature
Time (ms)
1000
100
10
1
hippooath.txt
Confucius_The_Great_Learning.txt
Apology_Plato.txt
null
StringReverseByLex
StringByLex

Assignment 7

- Red-Black Tree implementation for MyMap<K,V>
- Due: Friday, February 28, 2014 at 11:59pm

Midterm

- Next class - Tuesday, February 18, 2014
- Format
  - Multiple choice
  - Fill in the blank
  - Short answer
  - Recipe implementation
  - Test cases

Description	Operation & Input	Expected Output

Description	Operation & Input	Expected Output
Testing the size method for an empty MyMap	MyMap.empty().size()	0

Midterm Outline

1. Reuse
2. Linux/UNIX Review—command line
3. Test First/Test-Driven
4. Information hiding/access modifiers
5. Immutable vs. Mutable
6. Abstract data type (ADT)
7. Abstract class vs. Concrete class. Abstract method vs Concrete method
8. Java Syntax, including Liskov - Chapter 2: Review of Objects in Java
9. Write-Compile-Execute
10. Static Methods vs. Dynamic Method
11. JUnit testing
12. Designing test harness for given specifications
13. Abstraction barrier
14. Recipe for implementing an immutable ADT that is specified by an algebraic specification
15. Software process
16. Testing, including Liskov - Chapter 10: Testing and Debugging
17. Black-box testing vs. White-box testing
18. Dynamic Dispatch
19. Interchangeable Parts
20. Liskov - Chapter 1: Introduction
21. Debugging, including Liskov - Chapter 10: Testing and Debugging
22. The new rule
23. Factory method pattern
24. Effective Java items
25. Liskov - Chapter 3: Procedural Abstraction
26. Liskov - Chapter 4: Exceptions
27. Data Structures—List, Set, Map, Queue, Stack
28. Generics
29. Liskov - Chapter 5: Data Abstraction
30. Iterators, including Liskov - Chapter 6: Iteration Abstraction
31. Abstraction Function
32. Rep Invariant
33. Binary Search
34. Total Order
35. Binary Search Tree (BST)
36. Comparator
37. Asymptotic notation
38. Efficiency
39. Optimization