

Rebecca Rodriguez
CS 2420
Assignment 3
Analysis Document
2/5/15

Pair Programming and Partner Evaluation

I partnered with Steve Corey for this assignment. This is our second time working together, and we were just as productive as last time. We both deepened our appreciation of the skills the other brings to the partnership; as a result, we communicated and solved problems more efficiently. We both equally took turns as navigator and driver, and even came to recognize when the other was either in need of a break or was ready to rock and roll! I was pleased with the frequency of navigating and driving we each did. I was also quite pleased with our ability to recognize mistakes or problems, and quickly formulate an effective solution. I say quickly quite loosely, as it took us approximately twenty hours to finish the assignment. Regardless of time spent, I would most definitely work with Steve in the future. (Note: Steve turned in the source code.)

List vs. Array

Had we backed the sorted set with a Java List instead of a basic array, the implementation of MySortedSet class would have differed in the following ways:

1. A list allows for precise control over the insertion and removal of elements into and from the list, whereas an array requires extensive manipulation (i.e. copying, shifting, creating new arrays) when inserting and removing elements.
2. A list implements a special ListIterator, therefore creation of an iterator class would have been unnecessary.
3. A list has a generic type parameter, thereby making the backing collection of matching generic type. The array did not allow for generic types, therefore it was simply declared as an Object array.
4. A list provides searching methods to search for a given element; arrays do not. The contains method would have simply been a call to this method, rather than a method which uses binary search.

Using a list would have provided succinct method definitions, thereby reducing development time. However, because searching would be the most common operation on the elements in the set, implementing a List would be less efficient than an array in terms of running time. According to the Java API of a List, the search functions within List typically perform linear searches. Alternatively, our contains method performed logarithmic searches, which are much more efficient.

MySortedSet Contains Method

The expected Big-O behavior of MySortedSet's contains method is $O(\log(n))$. This is due to the use of the binary search algorithm in its definition.



Figure 1.1

In an experiment involving sets ranging from size 100,000 to 2,000,000, we tracked the running times of the contains method. After performing 10,000 iterations of the method on each different-sized set, the running time of the method averaged out to be approximately $O(\log(n))$. The logarithmic nature of the method is due to its use of the binary search.

MySortedSet Add Method

Performing similar test above on the add method, we calculated its running time to be approximately $O(\log(n))$, and $O(n\log(n))$ for worst cases. Because the add method calls the contains method, the binary search necessitates the logarithmic complexity. However, in worst cases, the backing array must be increased and all items are copied, thereby resulting in a linearithmic complexity.