Anthony Wilkinson – Analysis Document

1. Enrique Ramirez-Holston. He submitted the source code of our program.

2. We switched about once every 30-45 minutes. To me, this amount was acceptable, as we usually worked in sections of about 2 hours or so.

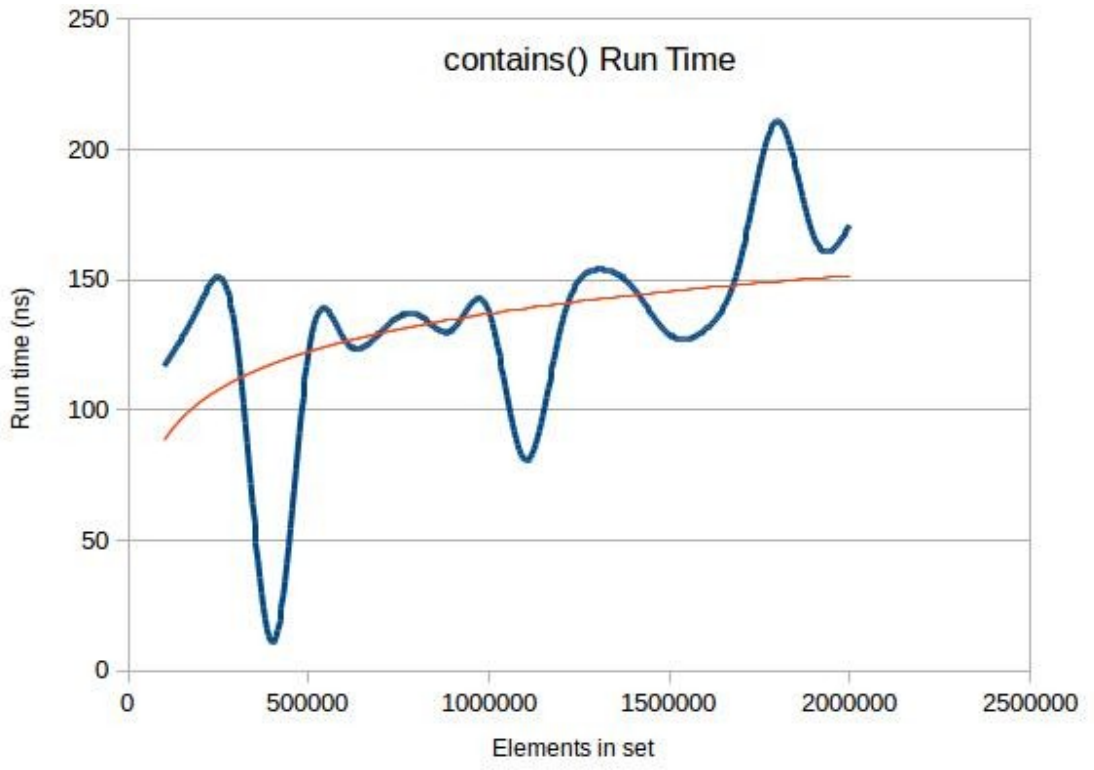
3. He's really good at picking out mistakes! He found flaws in our program multiple times that I might not have found until later! I plan on working with him for this next project, and then switching after that.

4. Main differences:
- Wouldn't have needed a “grow” section in the add method
- Might have used the List's built-in contains method rather than our own.
- Would have used List.get(index) rather than array access, which might have been slightly messier.

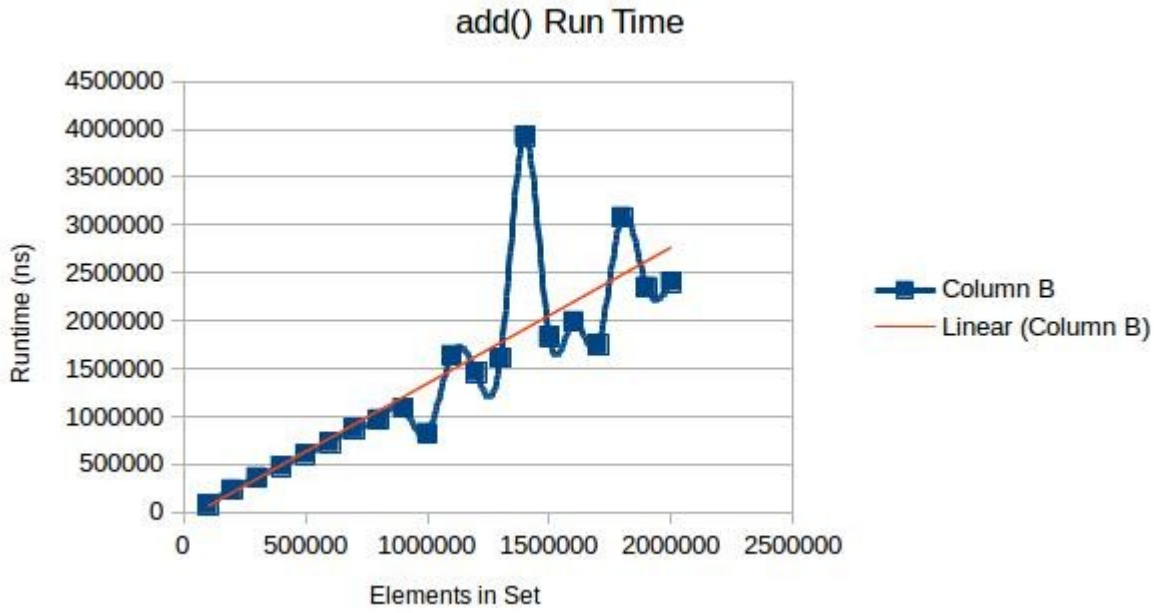
I suspect that using a List would be more efficient, simply because the writers of Java are probably more knowledgable about what it takes to achieve efficient growth. I also suspect that a List would have been more efficient in program development time, because we would no longer have needed to implement a “grow” section in our add method.

5. I would expect it to be O(log N), because binary search is a logarithmic algorithm (i.e. for every doubling of N, only one more operation would be needed).

6. The growth rate roughly matches O(log N) if we look at the red logarithmic trend line, but the results are not extremely accurate as far as the main blue line graph.



7. In the worst case, to **find the location** to add the element at is O(log N), as the add method utilizes a binary search to find where to put the element. On the other hand, **inserting** the element has a worst case O(N) performance because if an element is added at the beginning of the set, every element in the set will have to be moved up one position, giving N swaps.



8. In total, I spent over 10 hours on this assignment.