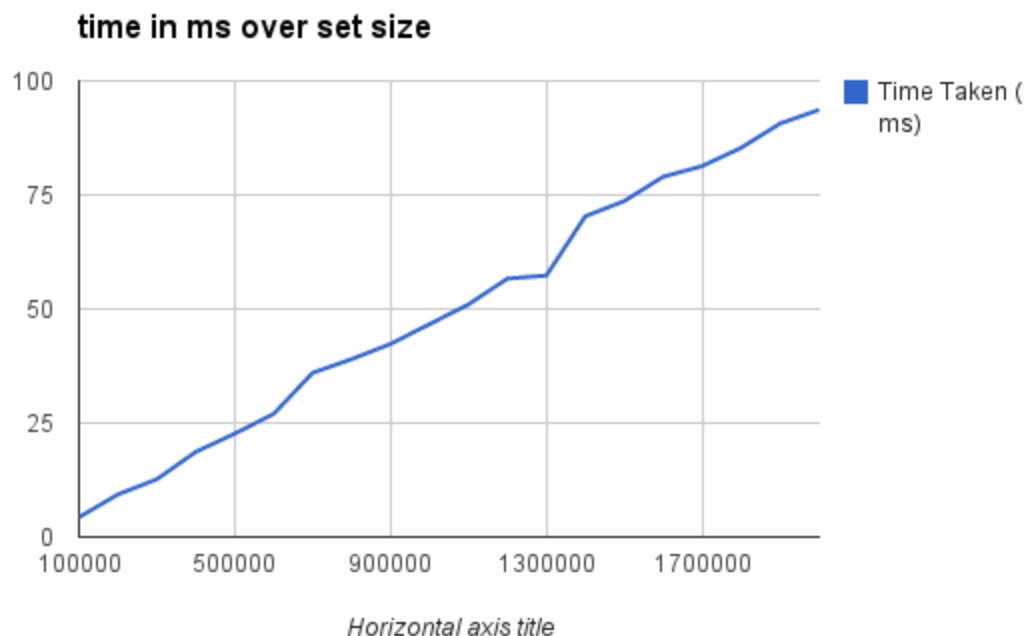


Alex Herrmann (u0913436)

Assignment 3 analysis

1. Partner was Nick Manuell, I submitted the code
2. Switched less times than last project, it ended up with me writing the code while nick tested what I had wrote and googled when we had problems.
3. I do! Nick's pretty cool.
4. The implementation would have been much much much simpler, even taking into account the binary search, a lot of our problems would have not even occurred with the shifting of all the data, and re-allocation and copying would not have come into play as it would have dynamically resized itself as we needed it.
5. $O(\log n)$, because that's what you told us... no, in all seriousness, the set of elements we have to search is halved for each time we search.
6. didn't get the $\log(n)$ performance we wanted, even after trying to optimize it before we simply ran out of time to tinker with it. I think it's probably in the cost of copying and doubling the size, maybe a larger data set would have prevented it.



7. After reading the question I'm really confused as to what you guys wanted, are you trying to benchmark the binary search function? Or testing the add function by adding and removing the

same number over and over again? The graph above is the data from adding so many integers generated with the Java RNG.

8. A lot, probably 12-15 all said and done if not more.