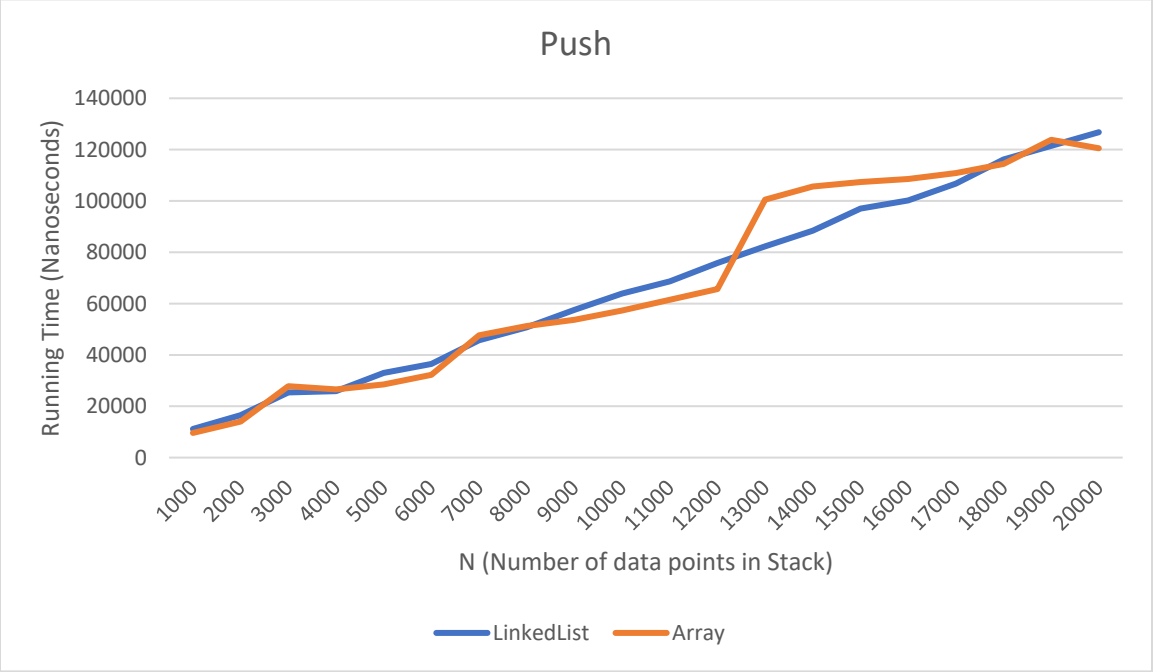
Analysis Document: Assign 6

1. I've spent around 20 hours coding and testing this assignment. I am submitting this code alone as I had significant difficulties with my partner for this assignment. It was our first assignment working together, and I ended up doing almost all (if not all) of the actual code on my own. He contributed to some of the tests and Javadoc comments, but only once I'd explicitly asked him to do so. I tried to ask him questions on how he thought we should implement the code, or opinions on how I should optimize the code I had just written, etc. but he didn't really give me thought-out answers, he usually just gave a vague answer or agreed with what I said. Additionally, while working on the code, he copied and pasted some old code from his old assignment as he had taken this class before, but did not receive a passing grade. I immediately told him I was not okay with him referencing his old code at all, especially since he admitted that his previous partner did most of the work. We had a discussion about academic misconduct and how to avoid it. I messaged the professors about this issue as I wanted to get guidance on how to avoid misconduct by him referencing old code, while still finding a way for my partner to contribute equally to future assignments. I messaged my partner to let him know that I had sent this message to the professors, and he said that regardless of whether or not he looked at his old code, he more than likely would not be contributing any more to the future assignments than he did on this one. I asked for clarification just to make sure I understood him correctly, that he wouldn't really be contributing to future assignments and I would end up doing the majority of the work. He replied and confirmed that yes, that is what he meant. Up until this point, I assumed that the reason he hadn't been contributing because he was trying to be cautious about avoiding misconduct especially since I had expressed my concerns with it. However, once he made it clear that he was intentionally not pulling his weight and wouldn't be doing so in the future, I messaged the professors discussing the messages (I attached photos of the messages in my Piazza post) and asked what I should do. They told me to submit the code on my own, and that Professor Kopta would talk to my partner. He also asked me to confirm that I did almost all (if not all) of the code on my own and my partner didn't do any of it. I explained that he did help with the testing and comments, but I did about 95% of the code on my own. I will be finding a new partner as Professor Kopta said I should, as I am not willing to continue to do all of the work for the assignments now that I know he will not be pulling his weight. I let my partner know about this, and he messaged back and said he believed it was a miscommunication. He said that he didn't know how I expected him to contribute to this assignment, especially since I took the lead at the beginning of the assignment to get us started. He also says that he didn't mean to insinuate that he would not contribute to future code. However, I don't feel that these statements are accurate or good reasoning. While it is true that I took the lead at the beginning of the assignment, the only reason I did so is because he was not speaking very much, so I decided that to help get us started and move past the awkward new partner phase, I would take the lead and open a conversation about how to create our code (which, as mentioned above, he didn't take a big part in). Additionally, I made a repeated, conscious effort throughout the entire process to ask him how he thought we should implement the methods, what he

thought we should do, etc. I made it very clear how he could contribute to the code and gave him plenty of opportunities to do so. Beyond that, I also don't feel that it should even be my responsibility to tell him how to contribute to the code, as this is a college level course, and he should know that it is expected for him to pull his weight and that he needs to actively contribute to the code. I especially feel he should have known how to contribute considering he has both taken this class before and also had a partner for the four previous assignments, and thus has had plenty of opportunity to know how to contribute to code. To my knowledge, I did not make it uncomfortable for him to code, and actively encouraged it the entire time, as well. Finally, I don't feel that it is an accurate representation to say that he didn't mean to insinuate he wouldn't contribute to the code, as he stated it clearly in one of his messages to me, and further confirmed that's what he meant when I asked him to clarify to make sure I understood him right. I did my absolute best to communicate clearly the whole time, and as stated above, made sure I clarified things just to double check I understood so I can't honestly say that miscommunication could have been the only issue here.

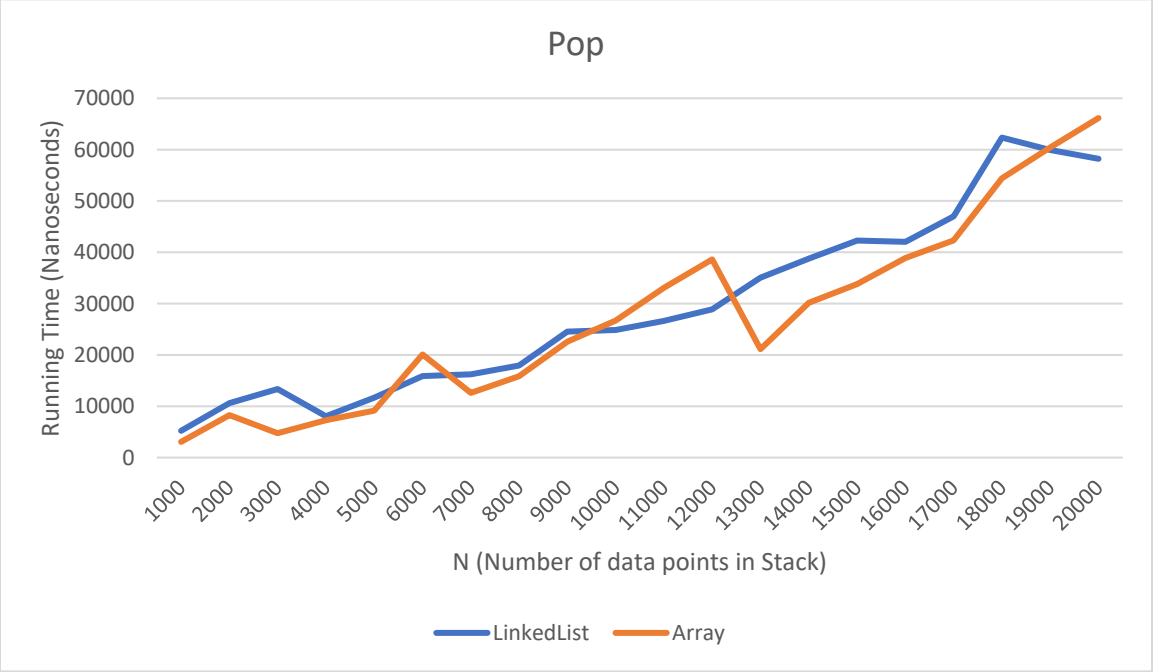
2. a.



The growth rate for both a LinkedListStack and an ArrayStack are both constant, or O(1). The timing method I used plots N * growth rate. The plot above is linear, as clearly seen from the graph, and also from the fact that the ratio between the actual running time divided by the expected running time stays roughly constant throughout the different problem sizes. Additionally, both types of stacks implement push() in a way such that it is not affected by the number of values in the stack; they instead keep track of the necessary values at all times and reassign them as needed, instead of looping through all the values in the stack just to add one to the beginning. This implementation produces a constant running time. The one exception to this

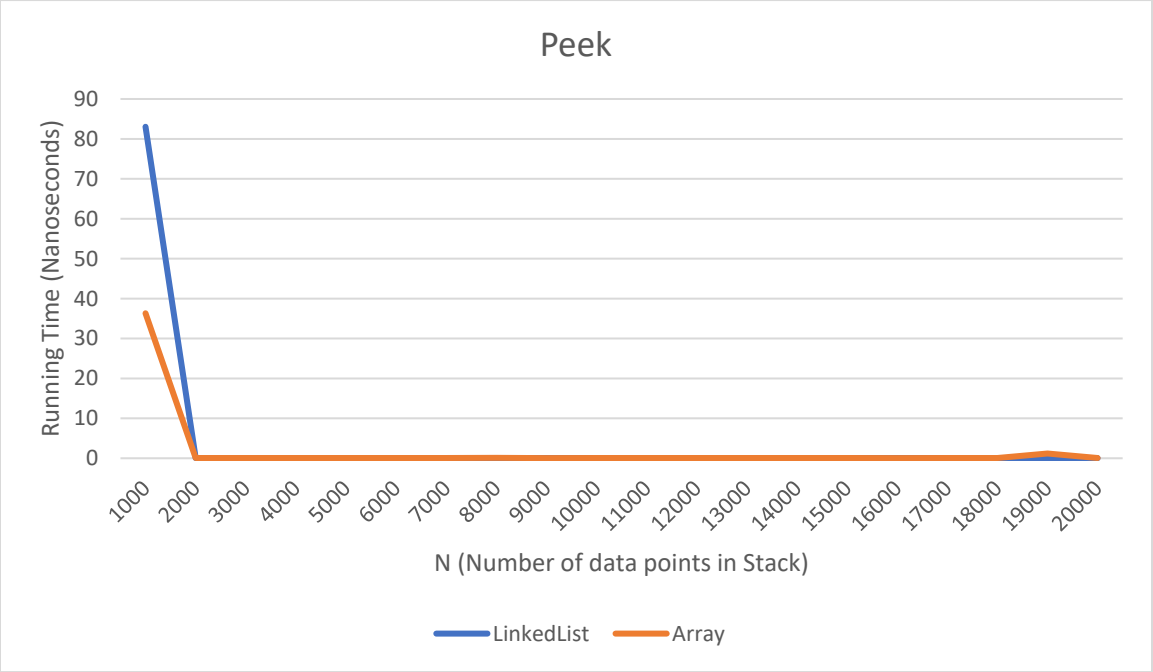
is the ArrayStack, however. If the array needs to grow, then it must loop through all values and copy them over to a new, larger array. This will cause the complexity to be O(N), making it slower than the LinkedList stack. However, in general, the two different stack types have about the same running time. Obviously, there are a few problem sizes that produced a slower running time for Array rather than LinkedList, as seen in the graph and as discussed above. However, because both push() methods have an average constant complexity, they are both fairly efficient. I would recommend using a LinkedList stack for this method, though.

2b.



Clearly, the pop() method had a lot more variation than the push() method. I believe this variation is actually because timing the pop() method requires us to also continually push values and then subtract the time it takes to do that, so that extra set of times could have made this plot less uniform. Both of these stack types have a constant complexity for pop(). The timing method for this was similar to above, where the plotted complexity is actually N * growth rate. In this case, it is clear from the plotted times and the relatively constant ratio between actual time/expected time that the pop() method is constant. The graph shows a linear complexity, and when calculating the ratio with N as the expected time, the ratio stays about the same (with some variation, but this is again most likely due to the unique way in which I had to time this method). It is also clear that both have a constant complexity from their implementation. Neither class uses any sort of for loop, while loop, etc. to pop the value. They simply keep track of the values as necessary and reassign them as needed, just like the push() method. This creates a constant complexity. In general, the time difference between the two stack types is negligible, because as discussed above, the variation between the two is likely due to the inaccuracy of the timing method, not an inefficiency in either type's method.

2c.



The growth rate for both types peek() method is constant. This is very clear from the graph, as it is an almost perfectly straight line after the first data point. I believe the discrepancy between the first data point and all the rest is simply due to the timer getting oriented, or the required computer processes to run the timing program in the beginning. If it is not already clear from the graph, when we look at the actual runtimes themselves, we see that they are almost all exactly the same, or at least within .05 nanoseconds of each other, which shows that the time it takes to run peek() is always the same, regardless of the number of data points in the stack. Additionally, the peek() method just returns the top value of the stack, which is something that both classes can access almost immediately. It requires no for-loops, while loops, etc. so it is a constant complexity, or O(1). You can use either type for this method, there is not one that runs faster than the other at almost any point.

3. I believe the LinkedListStack to be more efficient for the WebBrowser class, only because the push() method for ArrayStack has a worst case complexity of O(N), while LinkedListStack is always O(1). While ArrayStack is still pretty efficient, and in most cases will be about equivalent to LinkedListStack, an actual WebBrowser will be taking in large amounts of data and the O(N) complexity will cause a visible difference between the Array and LinkedList stacks.