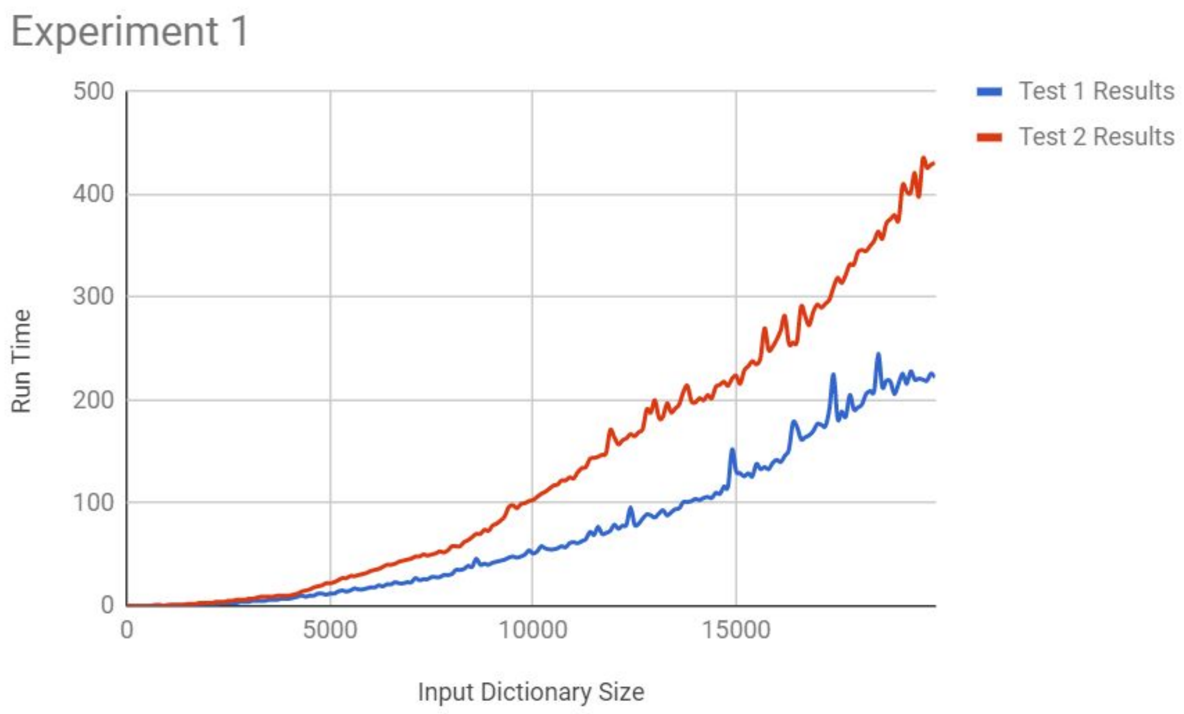
Kier Fisher, Zach McDaniel
Project 1 Group Write-up

1. When we needed to access the fields of an element we put an if statement to check if the element is null before looking at its fields. This stopped the program from throwing a null pointer exception. Also, this solution didn't take too much extra time as it was only a single if check every time the data was accessed.

Experiment 1)

A. Test 1 will return the average amount of time it takes to remove every element in the array from the first index of the Array Dictionary. Test 2 will return the average amount of time it takes to remove every element in the array from the last index of the Array Dictionary.
B. We predict that test 1 will be faster than test 2 because our remove method starts at the first index of the list and iterates to the end to find the item to be removed. Test 2 will have to iterate to the end of the list for every remove call while test 1 will only have to check the first value for every call. We predict that test 1 will have a linear curve and that test 2 will have an exponential curve.

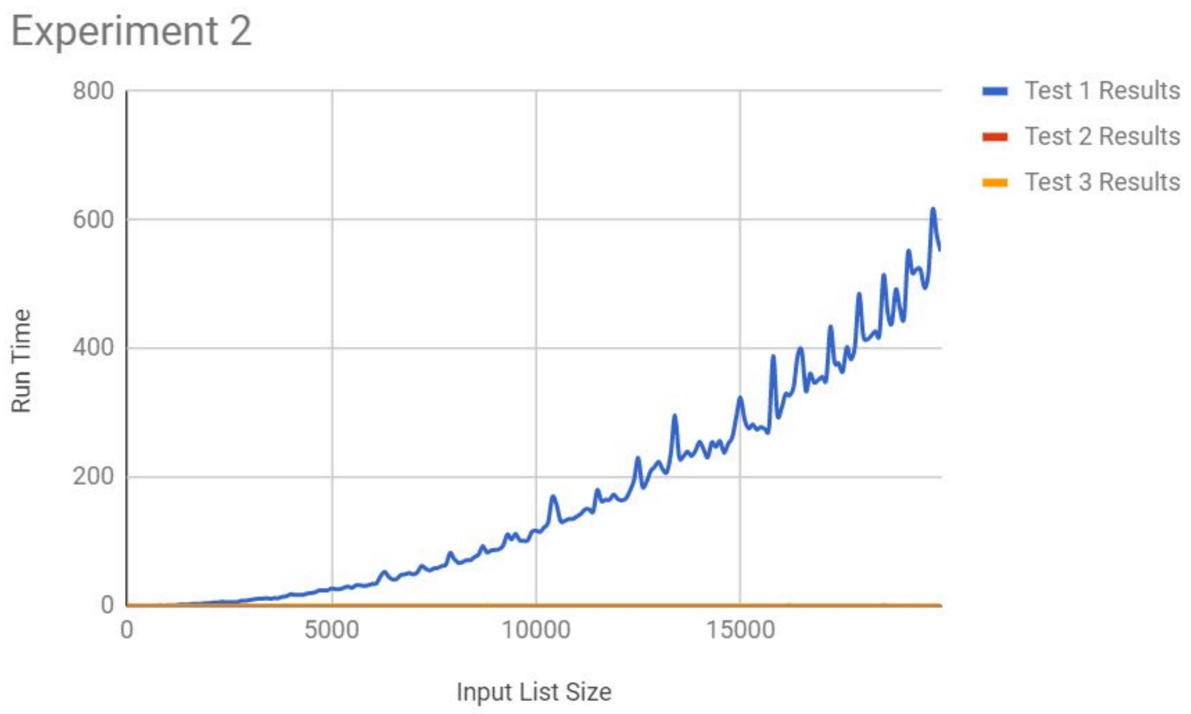


C.
D. Our predictions mostly fall in line with what the data shows. We predicted that test 1 would have a linear curve, but the data shows it to have a slight exponential curve. We think this is because in the remove method we start the index at 0 so we still have to iterate through the list. Test 2 takes longer because in test 1 when the remove method is

iterating through the list it only has to make a check to see that the value(s) at the front of the list is null (because it was removed) while test 2 also has to check if the keys match.

Experiment 2)

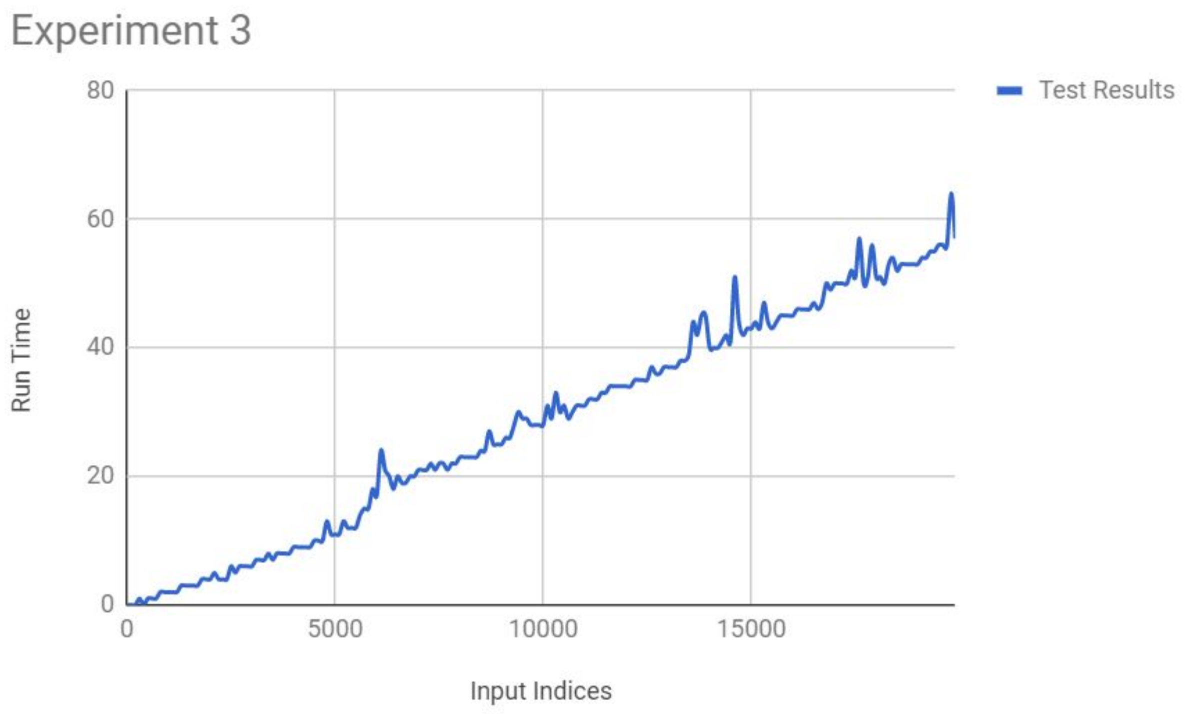
A. Test 1 returns the average amount of time it takes to get every element from the list. Test 2 returns the average amount of time it takes to iterate through the list directly using the list's iterator's hasNext method. Test 3 returns the average amount of time it takes to iterate through the list using a for each syntax.
B. We predict that that test 2 and test 3 will run in roughly the same time because both ways of iterating through the list happens in O(n) time using the iterator class. Test 1 will take longer to run because each time the get method is called it iterates through the list from front to back until it finds the value, so getting every element in the list will take O(n^2).



C.
D. Our predictions were in line with what the data shows. We were surprised about how much more efficient the iterator is versus the get method. We think this is because the iterator is only moving next pointers around versus the get method's if statement and for loop iteration.

Experiment 3)

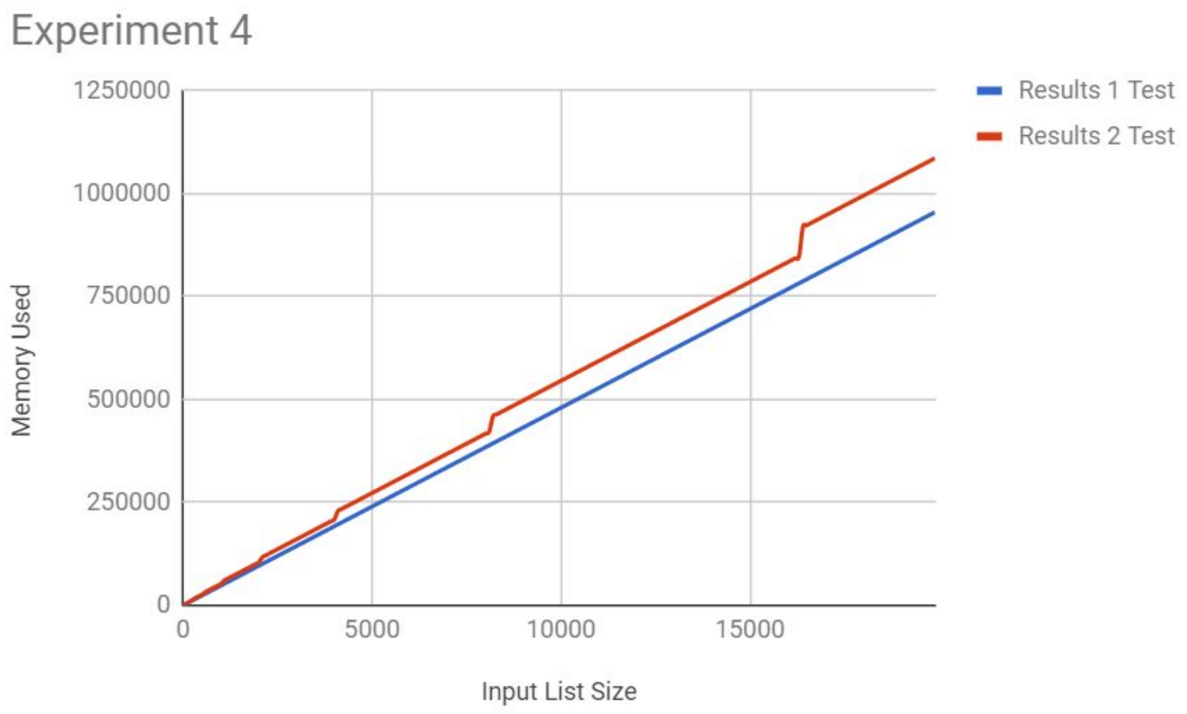
A. This test returns the average amount of times it takes to get every index in the array. It runs this test 1000 times for each array size and then takes the average time of those runs.
B. We predict that the time that it will take this test to run will grow linearly as the size of the list is directly proportional to the number of steps needed to get to the index at the end of the list. As the time to get the indexes is average getting indexes at the end of the array will bring up the average time.



C.
D. Our predictions were in line with what the data shows. Since the get method iterates through the list till it finds the given index it makes sense that as the index gets bigger the time increases linearly.

Experiment 4)

A. Experiment 4 is testing the amount of memory that it takes to store data in a Double Linked List and Array Dictionary. It does this by making a Double Linked List and Array Dictionary of different sizes and measuring how much memory it takes to store those data structures.
B. We predict that both graphs will have a linear curve because each increase in size corresponds to a proportional increase in data usage. We also predict that it will take more memory to store an Array Dictionary because it has more fields than a Double Linked List.



C.

D. Our prediction is in line with what the data shows. The Array dictionary consistently uses more data then the Double Linked List. We think the jumps in the graph for the Array Dictionary Test correspond to when the dictionary resizes its array.