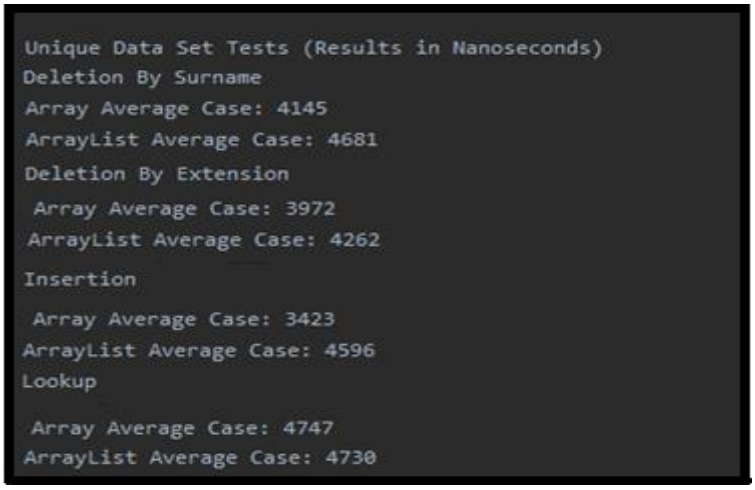
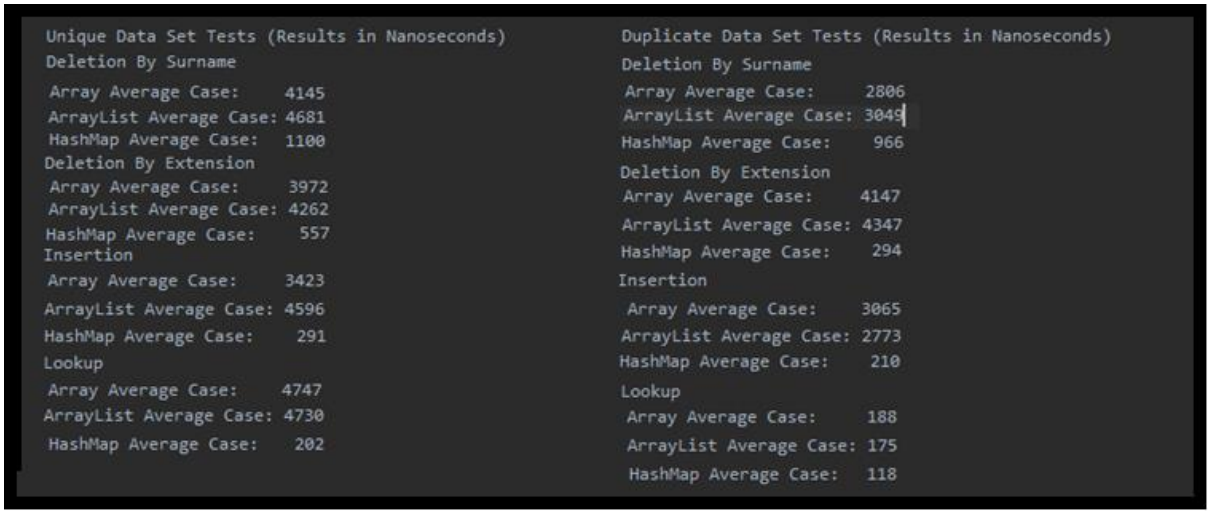
Performance Testing

Testing the same operations using the same data set on both the array and array-list directory illustrated that my array was ~7% to ~26% more efficient than my array-list on average, except for lookups where the array-list was ~3.5% better on average.



This could be because according to (Mathur, n.d.) in an array-list the references to the objects are not stored contiguously, therefore to iterate through an array-list of elements more memory must be traversed in order to find all elements – whereas in an array the stored items are stored contiguously therefore requiring less memory to be traversed as the elements are stored in the same section of memory, *(Thomas M, Et Al. 2017)* the contiguous allocation of elements in an array, results in less memory having to be traversed, my results reflect this as the average case of an array surpassed array-list in almost every test.



My results overall illustrated that the hash-map data structure out-performs both the array and array-list in *all* operations tested on both unique and duplicate data sets. I believe this is due to the manner in which elements of a hash map, are accessed, in comparison to an array or array-list. For instance - locating an object with a specific attribute for deletion of an entry in an array or array-list would require iterating through the entire collection of objects, to check each element for the specific attribute, furthermore this time to search increases as the number of elements increase. A hashmap in comparison uses a hash function to map each value to a given key, therefore one may pass a key and if the hashing is implemented correctly one may find out in constant time whether or not an object is mapped to that key. This explains why on average my hashmaps were beyond significantly efficient in contrast to array-list and array as elements can be accessed. (Mejia, 2019)

Correctness Testing

The below testing can be accessed via running the main method in the Testing class.

```
private static void validatingDirectoryInsertion(Directory directory){
    directory.insertEntry(new Entry( surname: "Taylor", initials: "L.T", extension: "00156"));
    directory.insertEntry(new Entry( surname: "Taylor", initials: "T.T", extension: "00088"));
    System.out.println("(HashMap) Size after two duplicate insertions: "+directory.toArrayList().size());
```

This method validates the behaviour of my system, is correct according to disallowing duplicate entries from affecting any given directory. The directory should not store both entries but instead only 1. Therefore, when testing this with all three directories the character “1”, should be outputted 3 times followed by a string.

```
(Array) Size after two duplicate insertions: 1
(ArrayList) Size after two duplicate insertions: 1
(HashMap) Size after two duplicate insertions: 1
```

When the following code is run with each directory the above is outputted therefore confirming that directories are not affected by duplicates.

```
private static void validateDirectoryLookup(Directory directory){
    System.out.println(directory.lookupExtension( surname: "Taylor"));
```

This method validates the behaviour of the lookup-method defined by the specification, given any directory this should output the extension corresponding to the entry with the specified surname. Using the previously tested array and array-list directory from the above test, and a new empty HashMap this should output “00156”, “00156” and null. In which it does. Therefore, also confirming the lookup method works.

```
00156
00156
null
```

References

Mathur, P. (n.d.). *Array vs ArrayList in Java - GeeksforGeeks*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/array-vs-arraylist-in-java/ [Accessed 21 Feb. 2020].

Thomas, M., Et Al. (2017). *A/AS Level Computer Science for WJEC/Eduqas Student Book (A Level Comp 2 Computer Science WJEC/Eduqas)*. Cambridge: Cambridge University Press,

Mejia, A. (2019). *Data Structures in JavaScript: Arrays, HashMaps, and Lists*. [online] Adrian Mejia Blog. Available at: https://adrianmejia.com/data-structures-time-complexity-for-beginners-arrays-hashmaps-linked-lists-stacks-queues-tutorial/ [Accessed 21 Feb. 2020].