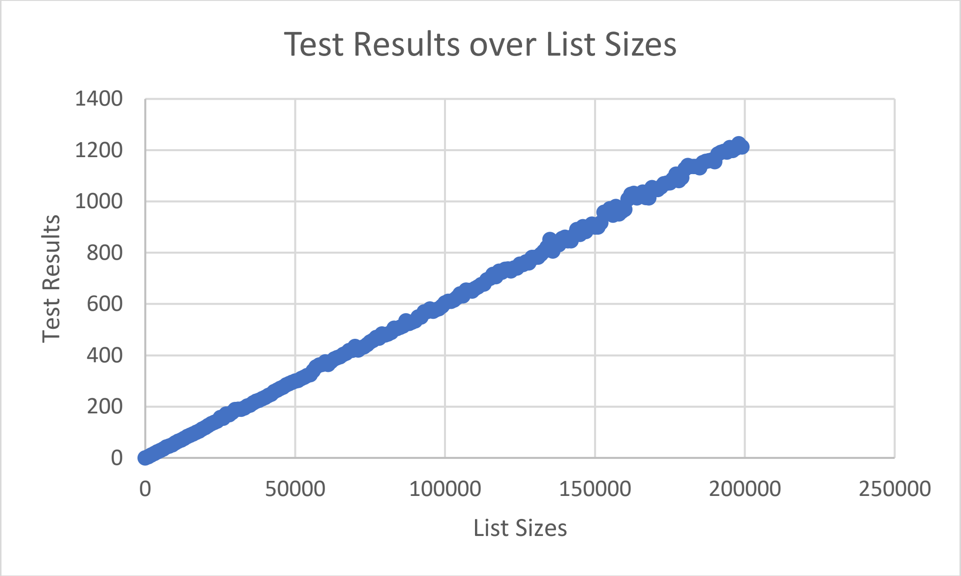
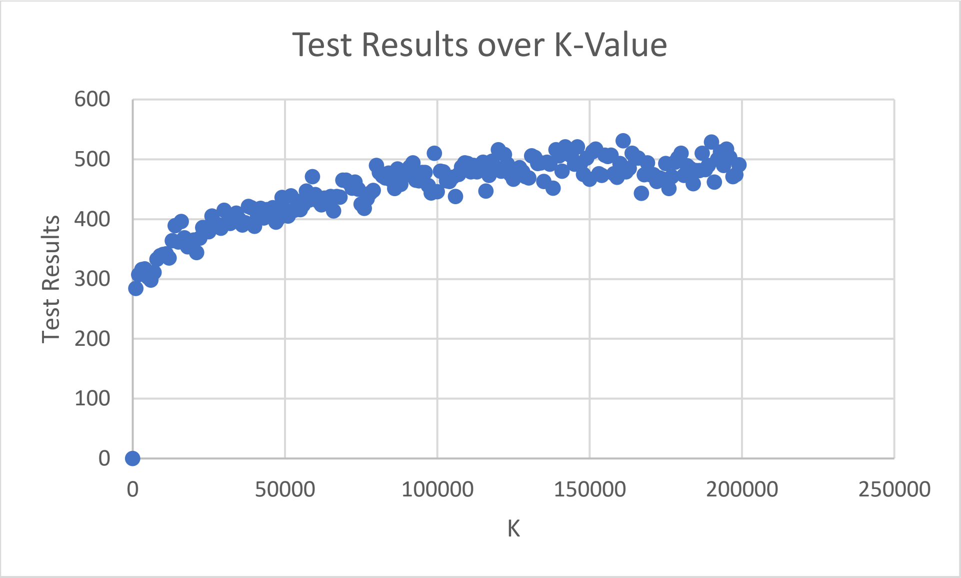
Part 1:

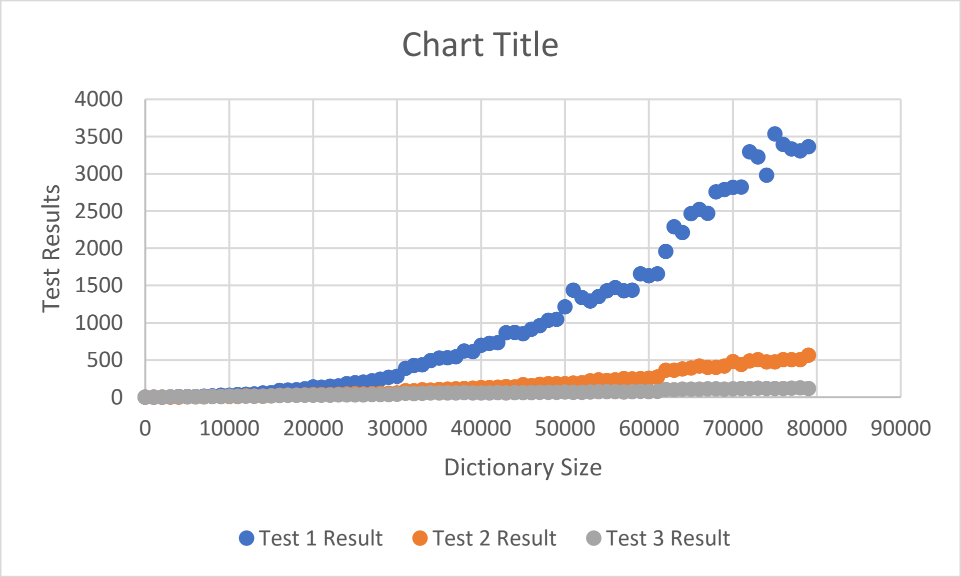
1. Does 5 runthroughs of the TopKSort method from the Searcher class on ILists of sizes of up to 200K, which I would say is quite the exceptional set of sizes, and logs the time taken.
2. Does 5 trials of the same TopKSort method on ILists of up to 200K size, and then logs the time taken.
3. This test is a doozy. It's first test builds a ChainedHashDictionary of size 200, and fills it with the contents of an IList, the keys being FakeString objects based on the contents, and then logs the time taken. The second test does the same thing effectively, although the keys are produced via a different FakeString method. The third test is the same, with another different FakeString method.

Part 2:

1. 



2.



3.

Part 3:

1. This result made sense, as the construction of keys and the getting of values all evaluated to about O(n) by my predictions
2. This test shows that there is a logarithmic relationship between k and the outcome.

3. These results shocked me mostly because the only difference was the Key generation methods, which appear to have different asymptotic runtimes, the first being somewhat O(n^2), the second O(nlog(n)), and the third about O(n).