# Evaluation

Average comparisons for sequential search on data 1 = 143

Average comparisons for binary search on data 1 = 12

Average comparisons for hashing search on data 1 = 10

Average comparisons for sequential search on data 2 = 1533

Average comparisons for binary search on data 2 = 19

Average comparisons for hashing search on data 2 = 3

Brief statements

1. Sequential search gets much worse as the size of the array gets larger, at 100 elements we see an average comparison of 143 & at 1000 elements we see that number jump to 1676.
2. We can see sequential search has a performance of an average O(N) with the number of average comparisons being roughly that of the number of elements within the array.
3. Binary search gets worse as the size of array gets larger too, though to a much lesser extent then sequential search having an average comparison of 12 at 100 elements & 31 at 1000. This seems to indicate that performance is quite stable with only 7 more comparisons for an array that is ten times larger than data 1.
4. Hashing performance, however seems to get better performance as the size of the array gets larger with an average of 10 comparisons at 100 data elements of an array size of 149 & only 3 at 1000 data elements of an array size 1499.
5. The performance of hashing is down to that perfect hashing only requires a single array access which has a performance of O(1).
6. It’s due to this best case why we see a better performance for data 2 which can only be due to that data 2 had less key collisions, then data 1 which could be down to the use of 1499 being the prime number used to get the hash as oppose to 151.