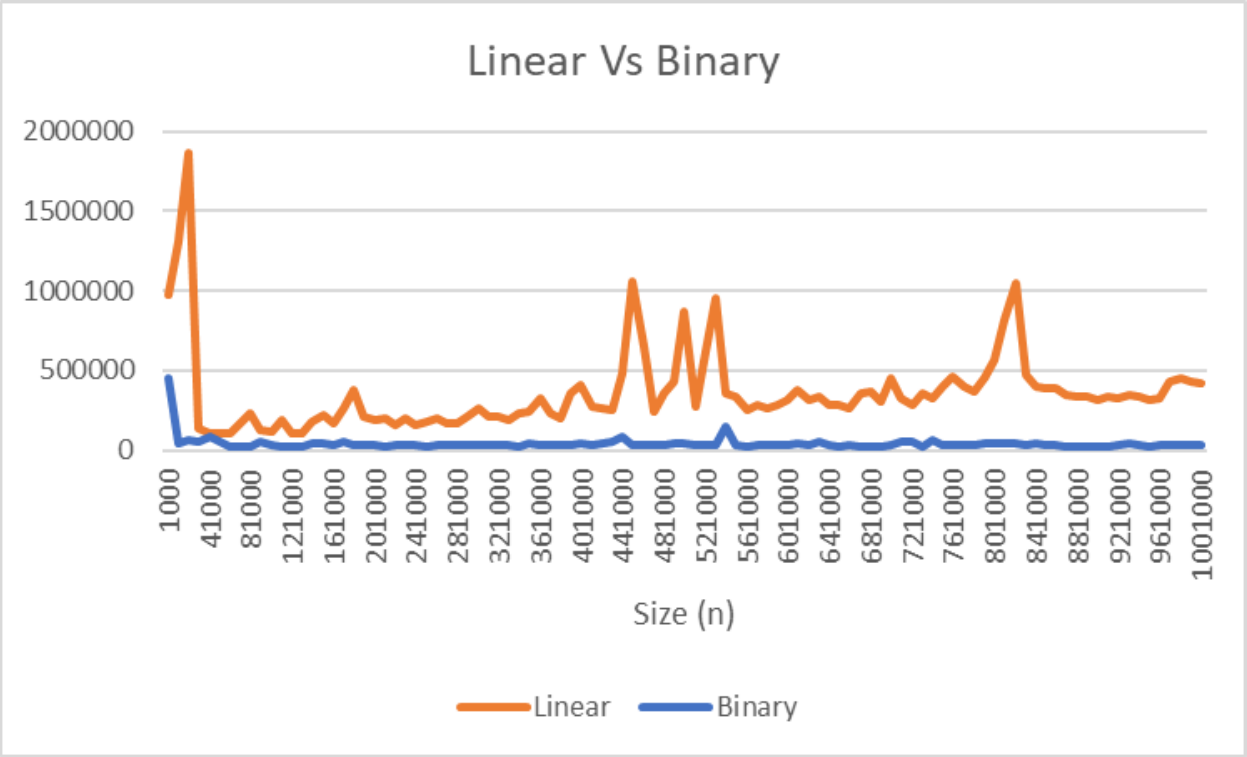
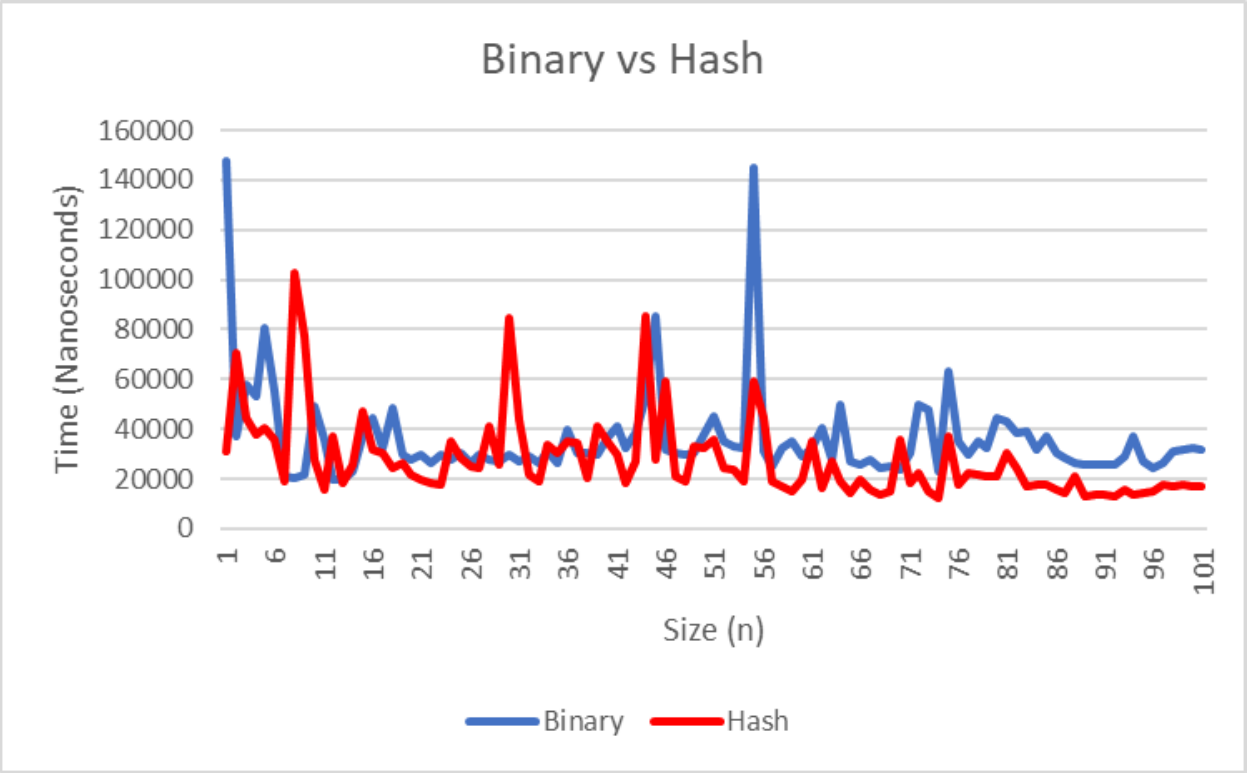


Part 1:

Regarding the graph for the comparison of linear search with binary search we can observe that the linear graph at all n values is slower than the binary search algorithm and the time it takes to find the elements increases linearly as the size n increases. The binary graph remains fairly constant at all sizes of n and is always faster than the linear search method. At the beginning of the execution at n of size 1000 for binary and 1000 to 4000 for linear the graphs start at a higher run time but after those two points the algorithms begin to run faster. Between these two methods we would choose the binary search method for all cases since judging from the graph, it is always faster and even taking sorting into account since quick sort is exceptionally fast it wouldn't make that much of a difference and binary would still be faster however other sorting methods may result in slower times.



Part 2:

Regarding the graph for the comparison of binary search with searching using a hashtable we can observe that the hash search is faster than the binary search for the majority of sizes of n. There is not a very big difference between the graphs. Since our graph reflects that hash searching is faster than binary we will choose that method over binary search. Since we are conducting multiple searches for different elements in the hash table the initial cost of creating the hash table is negligible and a hash search is still faster than a binary search in the long run.