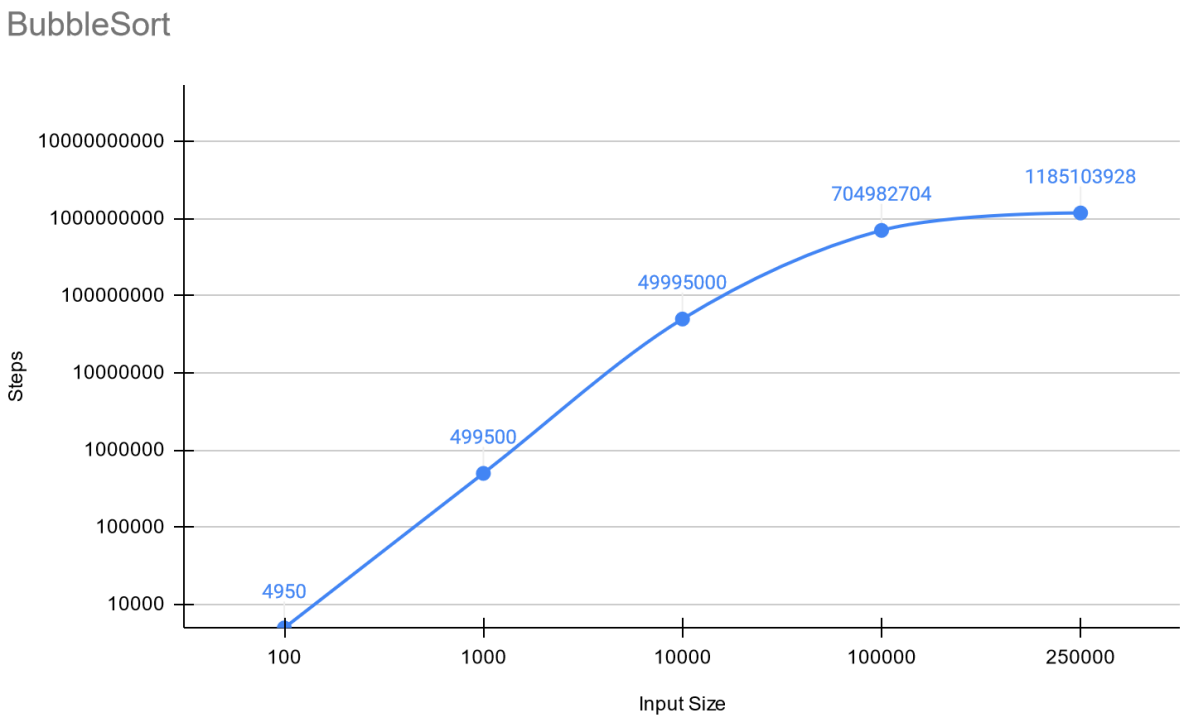
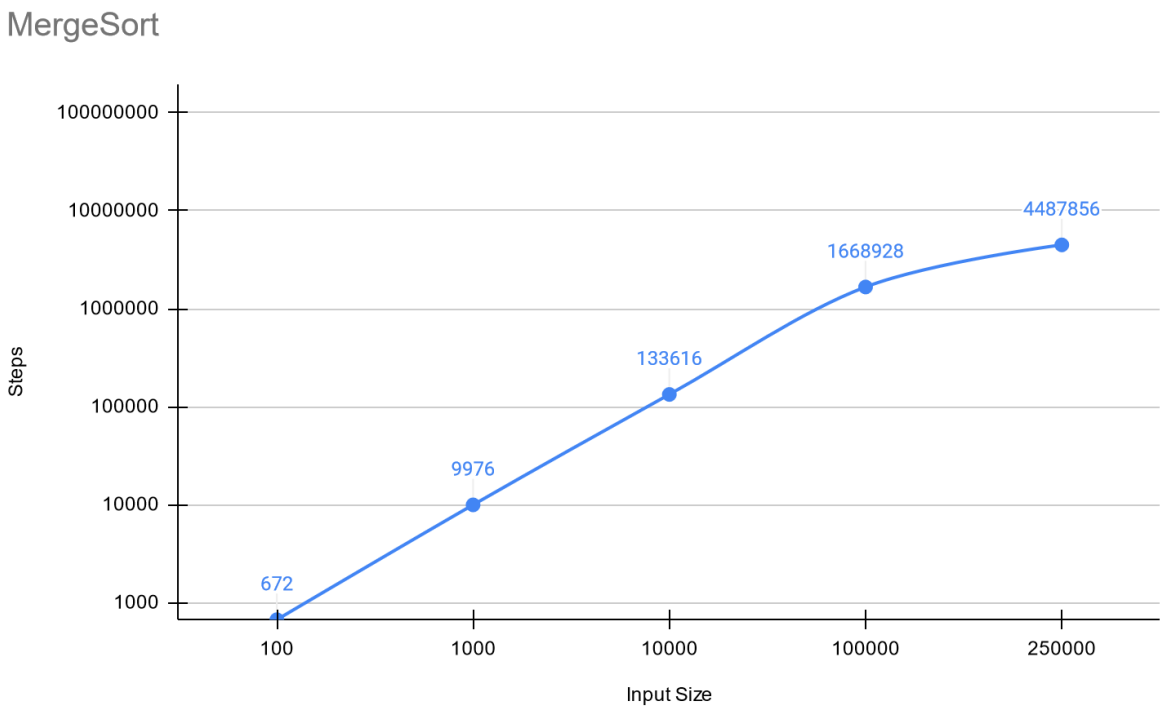
Conner McKevitt
Question 5

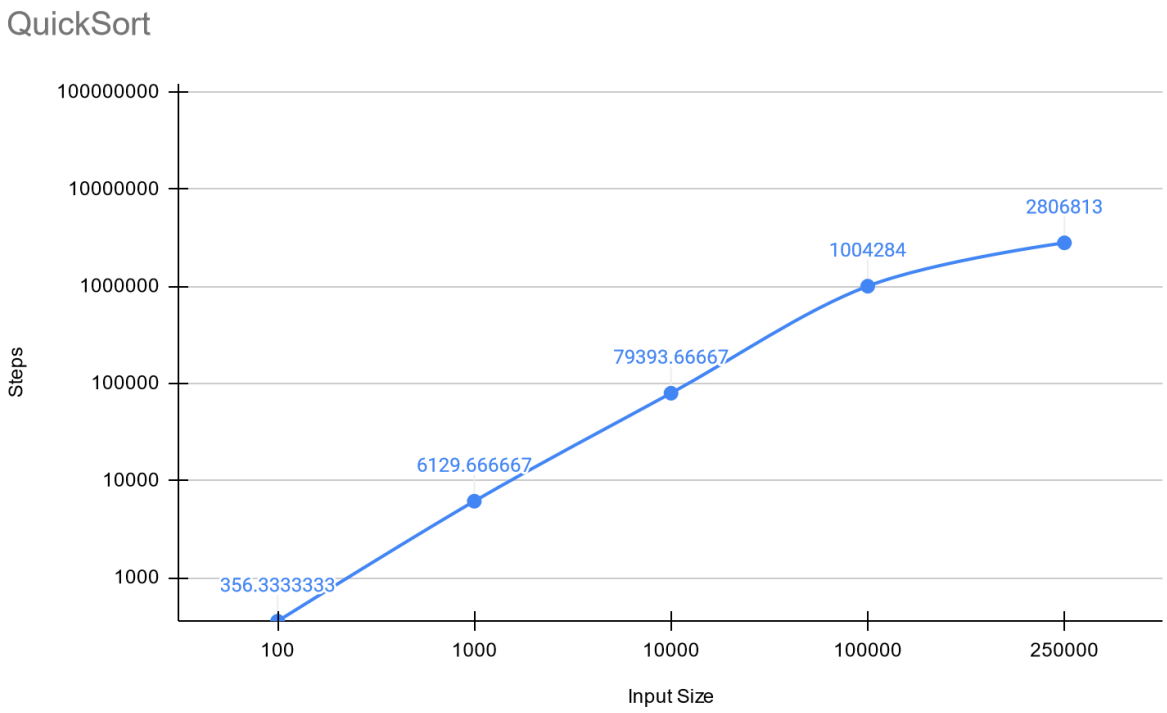
Each algorithm was done 3 times for each input size and the total steps for each input were then averaged. The range of numbers increased with the inputs size so for inputs size 1000 the range of numbers was 1-1000.



Bubble Sort sorts a list by comparing each adjacent item starting of list[0] and swapping the items if they are out of order. It does this for multiple iterations and the final result is a sorted list. Bubblesorts average and worst case time complexity are O(n^2)



Merge sort sorts by dividing the input into two halves then recursively calling itself on those two halves. Eventually the input will be divided into individual elements. Merge sort then repeatedly merges these until there is only one list, which will be sorted. Merge sort has an average nlogn time complexity



Quicksort divides the input by determining a pivot, in the algorithm I implemented I always chose the last element. It then reorders the list so that all values less than the pivote are before it and those larger are after. Then it recursively calls itself on the two sum list before and after the original pivot. The end result is a sorted list. Quicksort on average runs in O(nlogn) and a worse case of O(n^2).