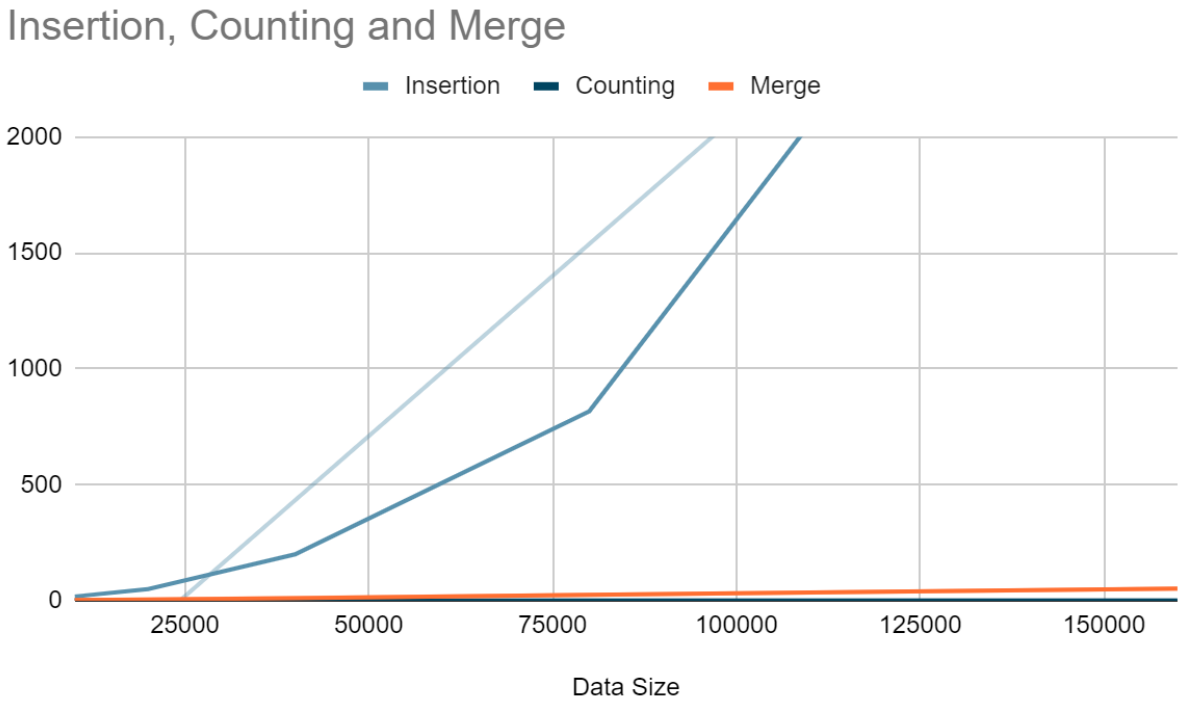
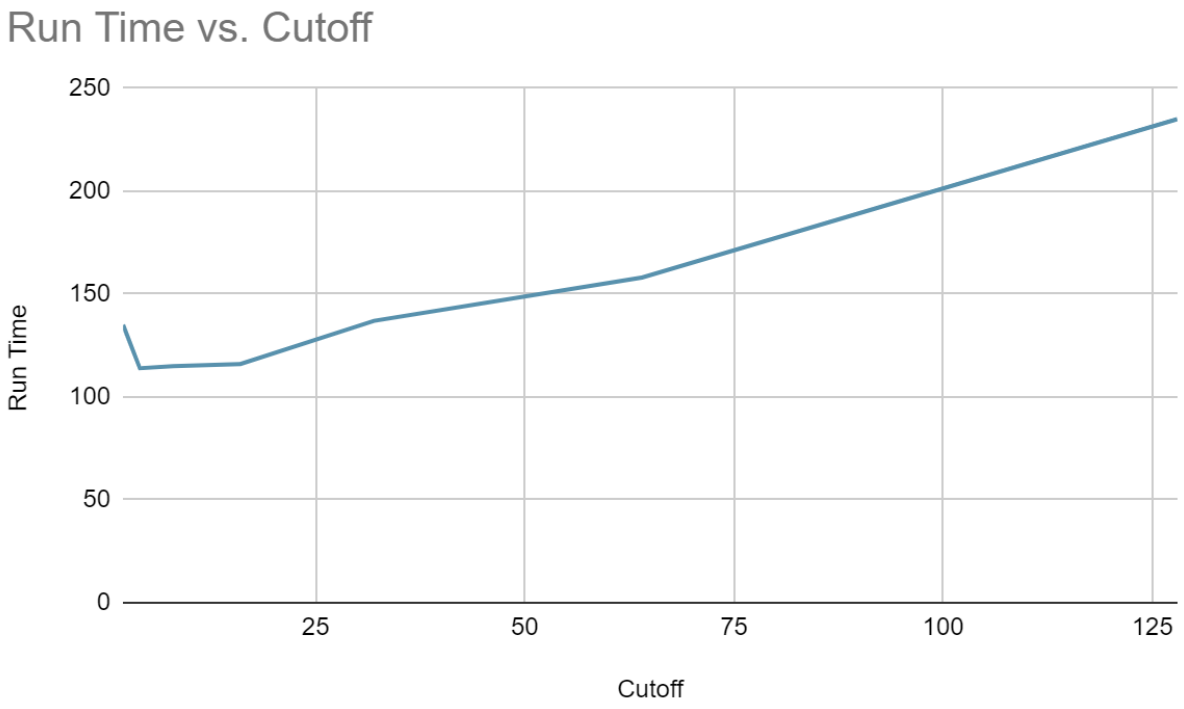
Task 1



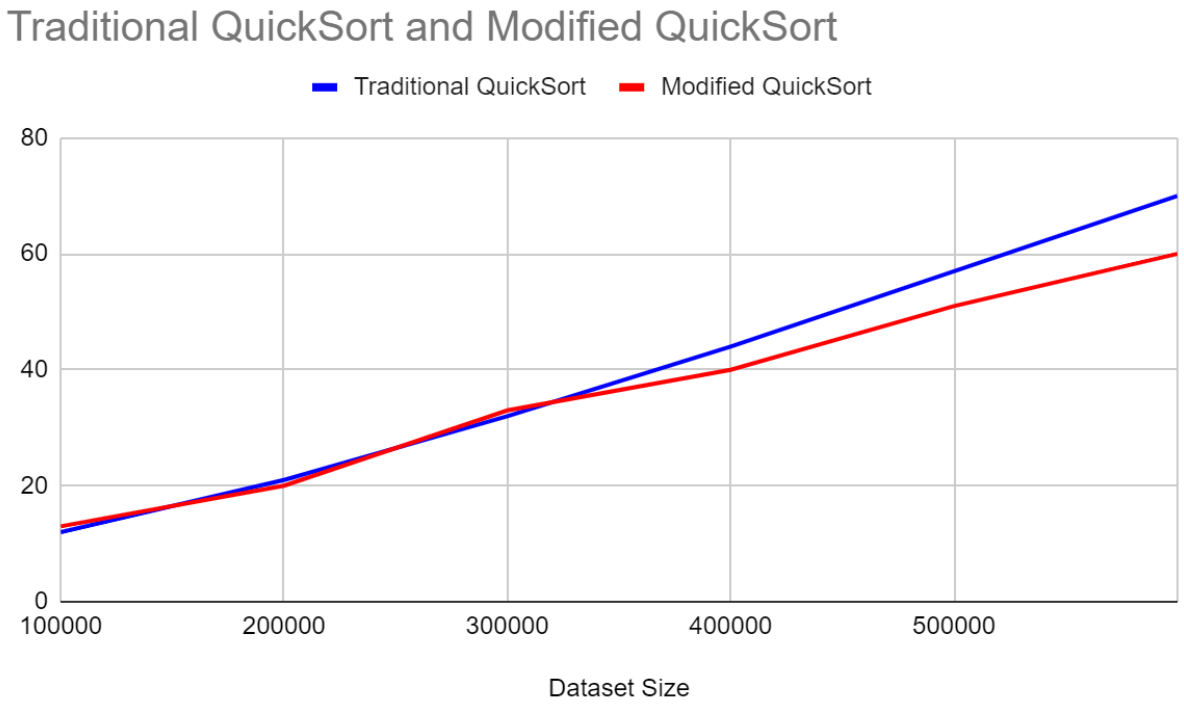
Conclusion:

The Insersion sort grow so much faster than the other two sorting method, since the complexity of Insersion sort is $O(n^2)$ and the complexity of counting $O(n)$ and merge sort is $O(nlog(n)$.

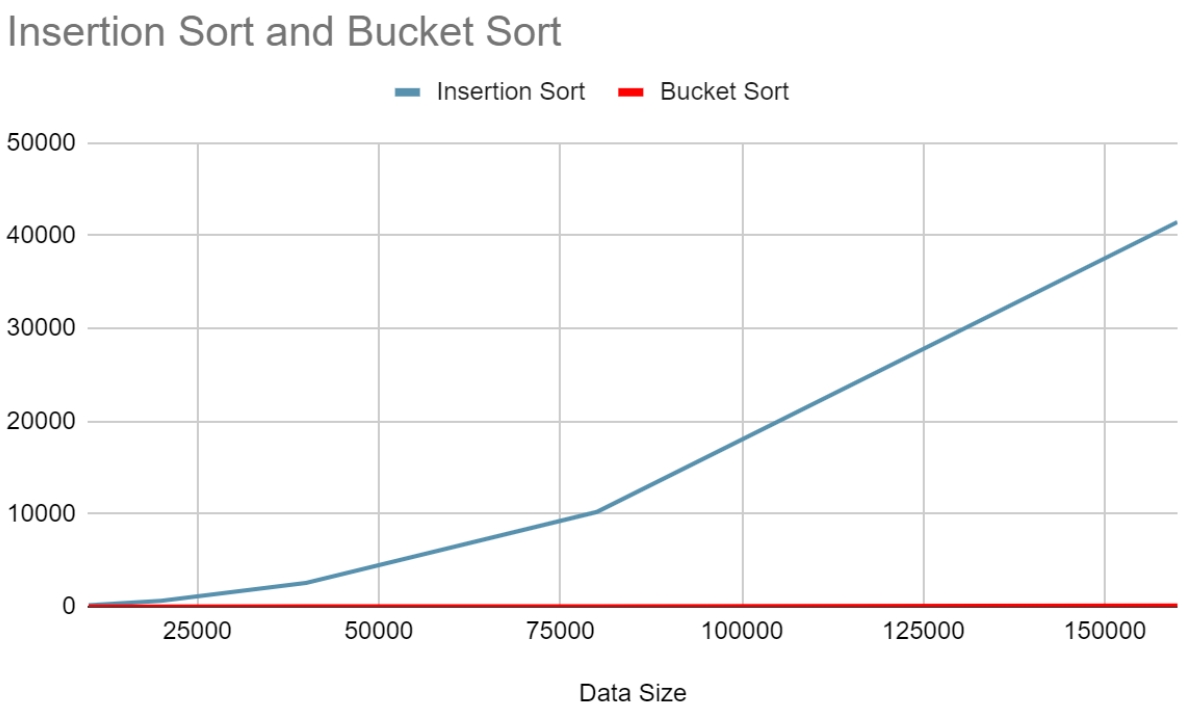
Task 2:



The best cutoff is at 4.
Task 3:

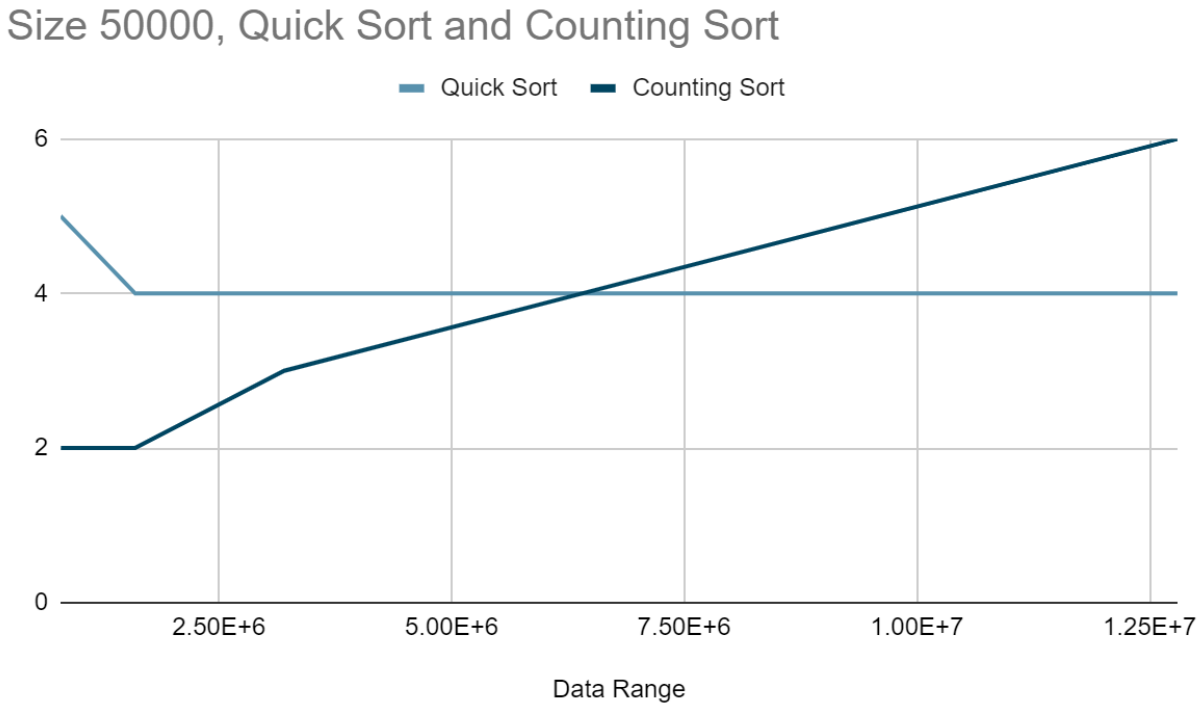


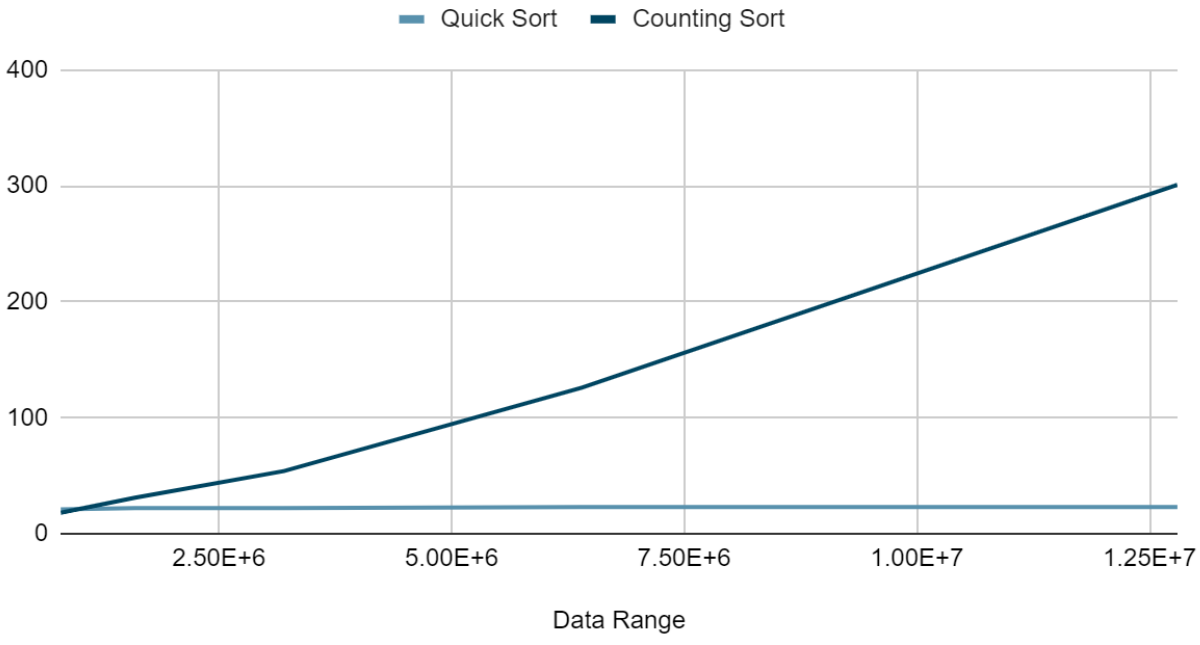
Task 4:



Since the complexity of insertion sort is O(n^2) which is so much greater than bucket sort(O(n)). When n increases, the blue line increments so much more than the red line.

Task 5:


Size 50000, Quick Sort and Counting Sort
Quick Sort
Counting Sort
6
4
2
0
2.50E+6
5.00E+6
7.50E+6
1.00E+7
1.25E+7
Data Range


Size 200000, Quick Sort and Counting Sort
Quick Sort
Counting Sort
400
300
200
100
0
2.50E+6
5.00E+6
7.50E+6
1.00E+7
1.25E+7
Data Range

The range of data is completely uncorrelated to the running time of quick sort, since it treats every element as a byte. However, the running time of counting sort increases dramatically as the range of data increases. It is because the counting sort needs to create a new array with length of the maximum value of array.