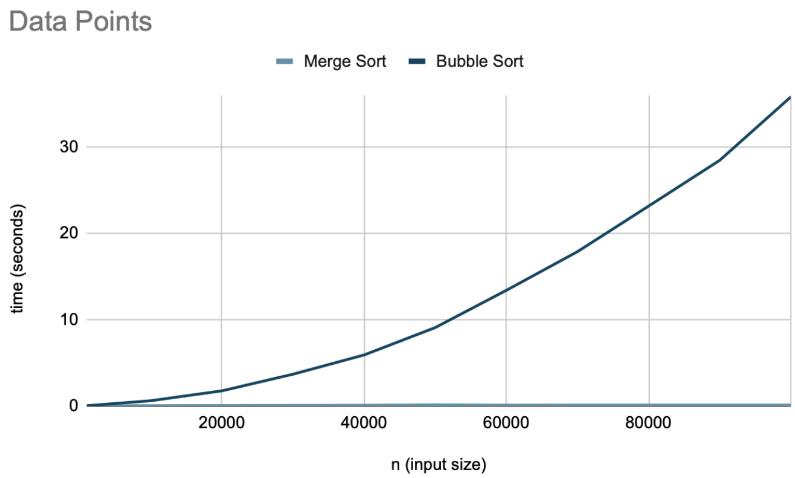

	Merge Sort	Bubble Sort
1000	0.012416669	0.024820014
10000	0.029710162	0.602095816
20000	0.047838508	1.747998642
30000	0.062429669	3.675540067
40000	0.080597402	5.905458108
50000	0.134090918	9.077735948
60000	0.096029407	13.39447895
70000	0.10484465	17.87273916
80000	0.11094068	23.15028187
90000	0.107481135	28.45430052
100000	0.110450346	35.79852062

As seen in the graph above, you can barely see a change in the mergesort line or time (which is at the bottom and looks almost like a horizontal line). The time for bubble sort increases at a large rate for each input size while as the input size of merge sort increases, there is barely an increase in the time in seconds. This shows how the big O notation of bubble sort represents quadratic growth or $O(n^2)$ since it is growing at a rapid rate and the worst case scenario for it is n^2. The time complexity represented by merge sort is O(logn) because it's worst case is logn. This is because the loop variable is being divided. This is clearly more efficient than bubble sort because no matter the input size, the time complexity will remain relatively low while with bubble sort, it increases dramatically.