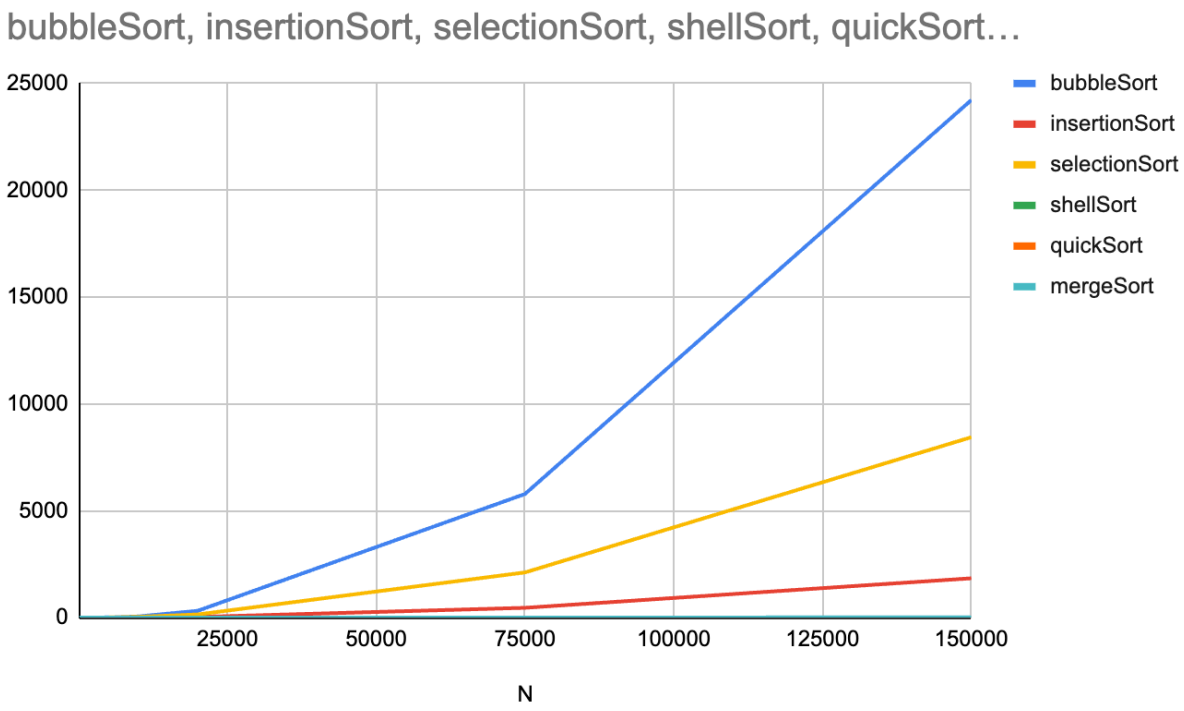

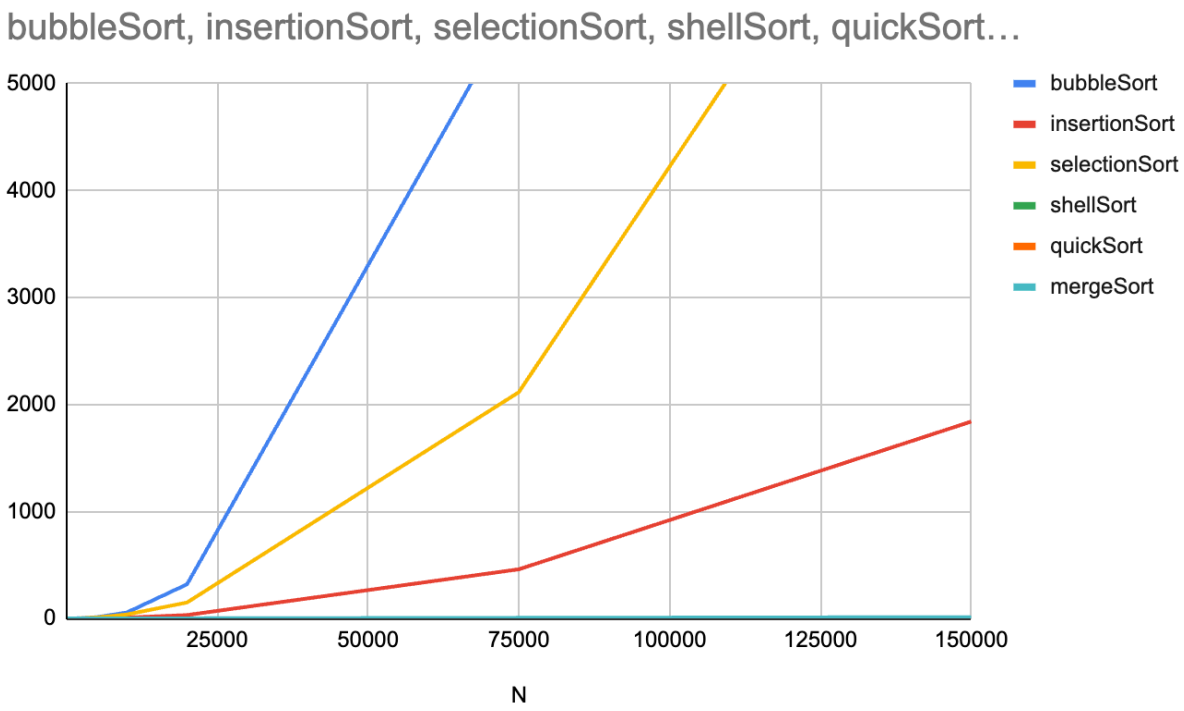

- Graph with lower y axis to see the quicker sorting closer

Shell sort and quick sort are very fast so they read as no time in milliseconds so they are along the x axis straight. Mergesort takes a very short amount of time but slightly over 0 ms so you can still see the line. I am not sure also what issue I was supposed to run into with graphing but I would think it's that you cannot see the very fast algorithms?

Bubble sort increases very quickly which means it's very slow. Second is selection sort followed by insertion sort. So bubble sort was in fact the slowest which was one of my predictions! I also knew this because I already ran my tests but it is hard to conceptualize just ms numbers. On this chart it is very clearly the slowest despite all three technically being O(n^2) runtime algorithms. Also once selection sort got to a significantly large sized array around 75000 it increased exponentially (to say a lot). Insertion sort did the same around 75000 but with a less extreme increase, which makes sense since it was already generally much faster than selection. Quick sort and shell sort are so fast that you can't even see them when mapped in milliseconds. I would have to convert it to probably nanoseconds to see it on a graph or maybe if i used an extremely large input size. However I think for mergesort the runtime complexity is always the same with no best/worst case it is just the theta case. Finally the merge sort was just nice and quick. It is in the middle so I don't have much to say. It also got slower pretty steadily, maybe linearly but the graph is too small for me to tell and I don't think that is right.