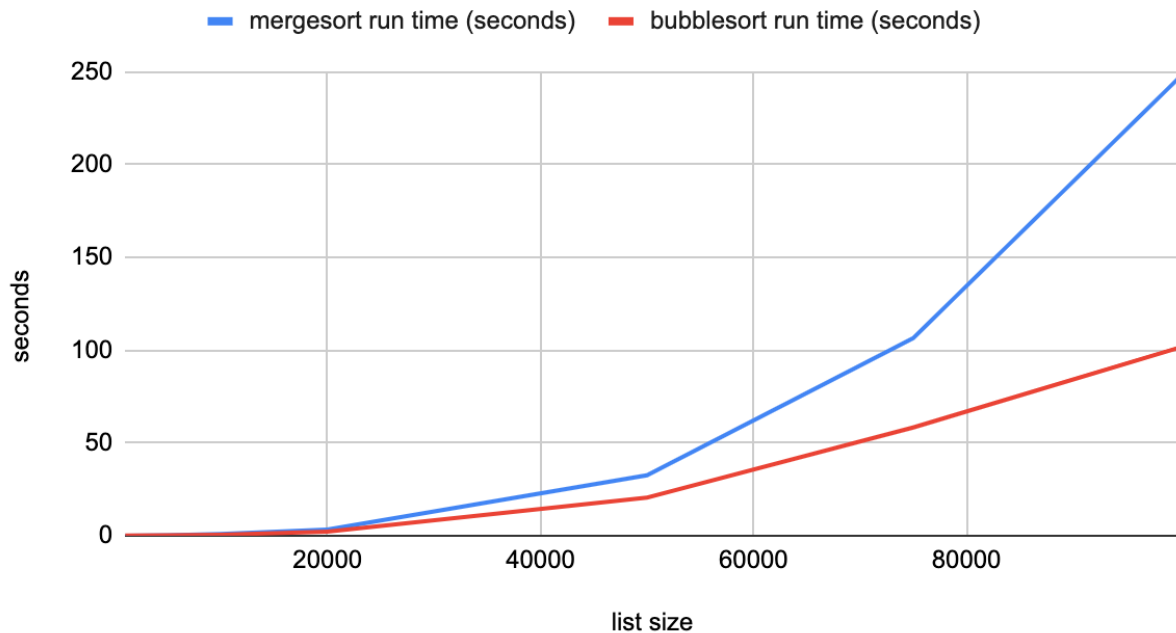


mergesort vs bubblesort run time



The graph shows pretty clearly that the bubblesort algorithm is more efficient than the mergesort algorithm. At the early data points, the two algorithms have pretty similar runtimes. As the size of the data increases, however, a large discrepancy begins to form. For the largest dataset, it takes mergesort more than twice as long to sort. This is because mergesort requires twice as much storage as the unsorted array since it makes a temporary array.

The time complexity of mergesort is $O(\log n * n)$ in all three cases; best, average, and worst. This is because the algorithm recursively divides the array in half, which is $O(\log n)$. Merging the arrays back together has a complexity of $O(n)$ so the final complexity is $O(\log * n)$.

The time complexity of bubblesort is $O(n^2)$. The best case time complexity is $O(n)$, when the array is already sorted.