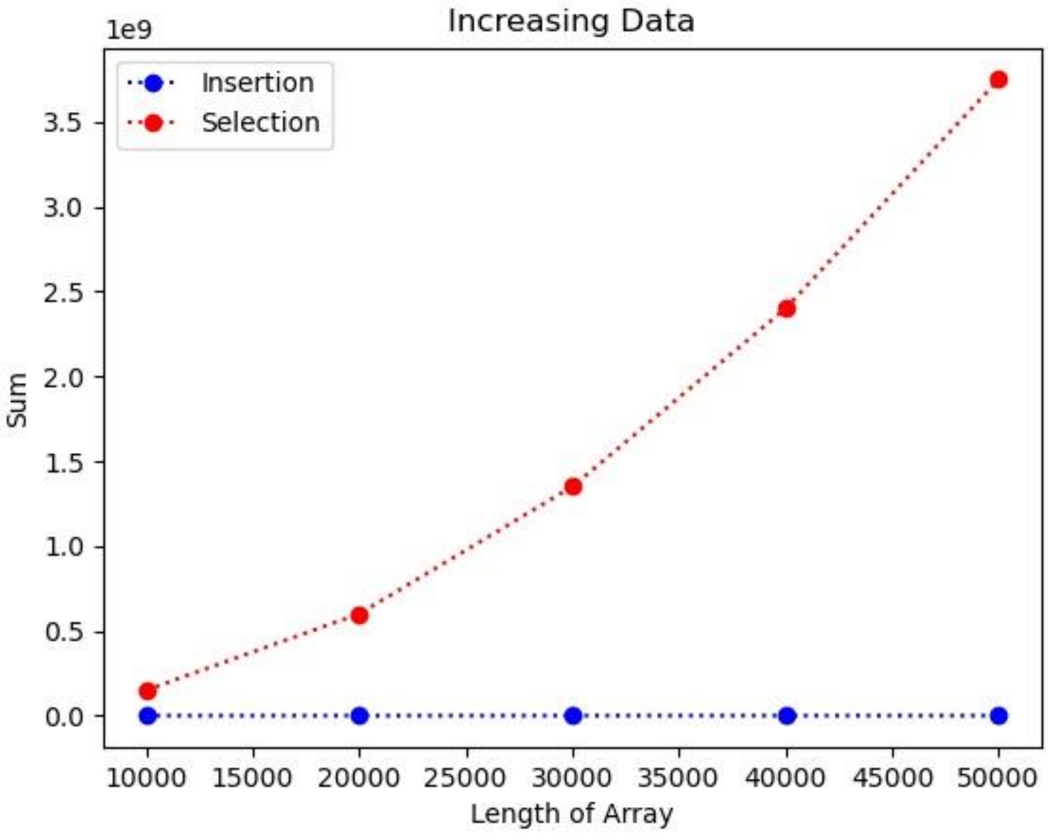
Initially Increasing Data



The insertion sort takes much fewer steps because for a list of increasing data, it is in its best-case scenario. This means that since the array is already sorted, our program can ignore much of the insertion sort code that deals with dealing with unordered data, and the insertion sort becomes extremely efficient for this case, hence resulting in almost linear graph for insertion sort. Meanwhile the selection sort doesn't change the way it runs regardless of how the data is ordered and although the array is sorted, it doesn't make the sort any faster, meaning that the selection sort always takes close to as long for the worst-case scenario, which is why the insertion sort becomes much more efficient for the best-case scenario of initially increasing data.