The insertion sort work by comparing one entry with the next entry in an array. Let's assume for all of these sorts, the entries are being sorted least to greatest. In insertion, if the next entry is less than the entry before it, then it will swap. Other than that, it will then start comparing the next two entries. The amount of time insertion sort takes roughly depends on if the entries are already sorted and how long the entry is. For example, the longer the entry gets, the longer the algorithm will take. This gives the algorithm an average running time of O(n^2). This algorithm can be faster if the entry is short and already somewhat sorted.

Quicksort is an algorithm that separates an array depending on a pivot element. The two arrays that are separated into are either lower or higher than the pivot element. It will then sort after it is separated by the pivot element. This sort can range anywhere from O(n)-O(n^2). On average the runtime for this algorithm is O(n log n)

Merge sort is an algorithm that separates the array into smaller arrays usually pairs. It will then compare the pairs, and group the smaller elements with each other. It will then sort the elements into a sorted array. The runtime for this algorithm can be anywhere from O(n)-O(n log n). On average the runtime for this algorithm will be O(n log n).

Gold’s Poresort works by splitting the algorithm into two phases. The algorithm then sorts the elements within the phases, then combines them at the end. The runtime for this sort, on average, is O(n^2). The runtime for this sort can range anywhere from O(n)-O(n^2).