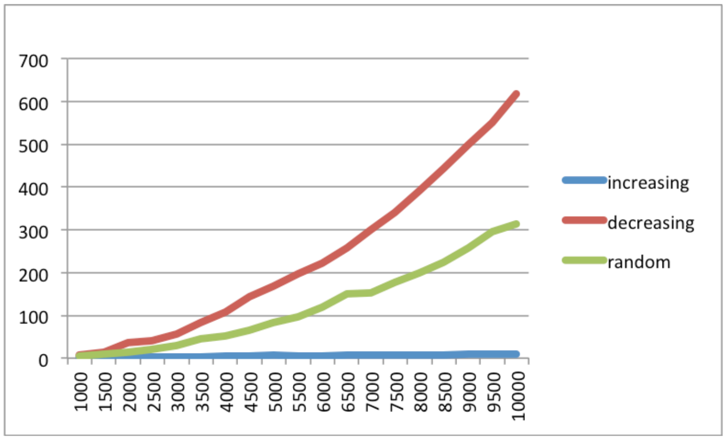
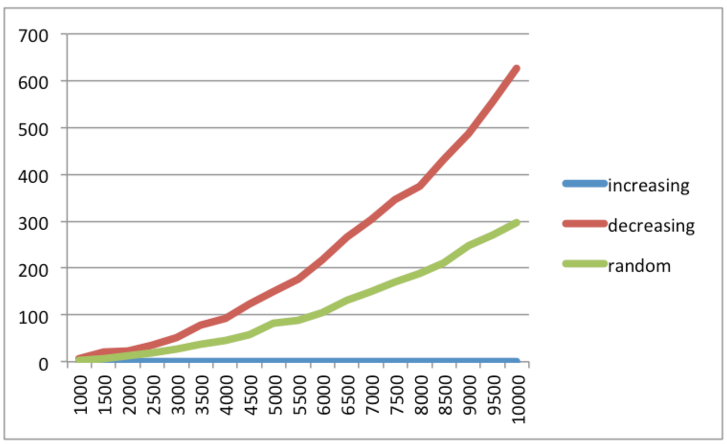
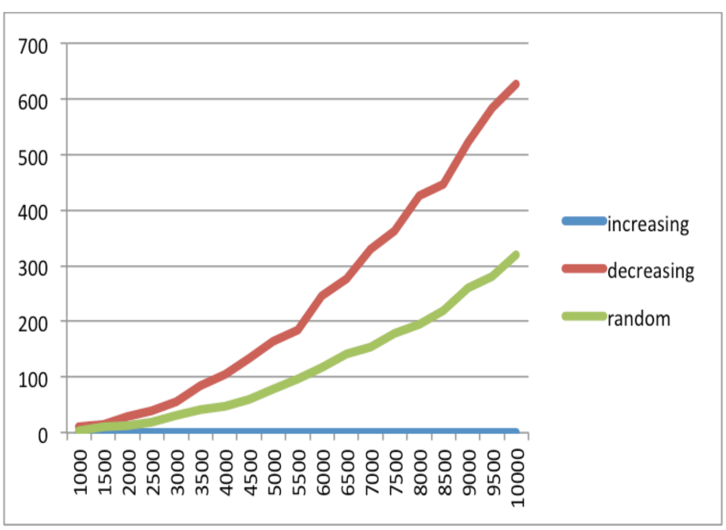
Insertion Sort



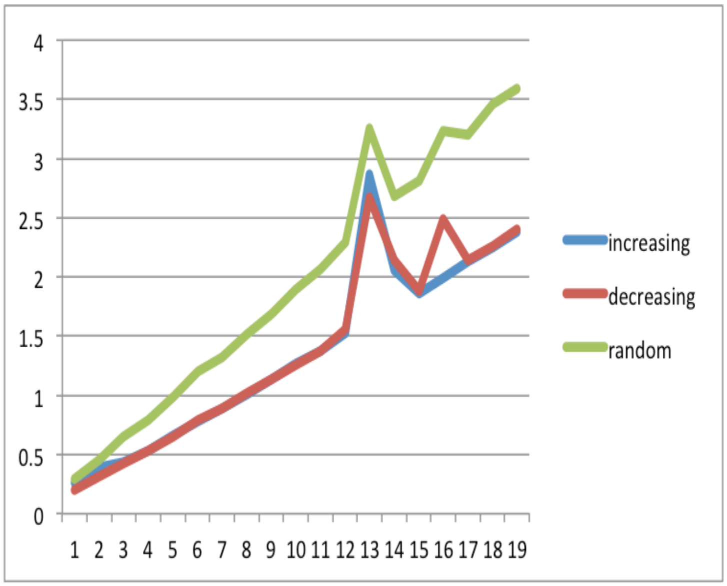
Bubble Sort



Quick Sort



Merge Sort



Merge sort worst case is the same as its best case which is O(nlogn). Insertion sort best case is when the input is an array that is already sorted. In this case insertion sort has a linear running time. Worst case is for insertion sort is O(n^2) because it must go through every iteration. Quick sort worst case is O(n^2), although this behavior is rare. Best case is for quick sort is O(nlogn). Choice of pivot in quick sort can drastically affect the running time since it will perform better for increasing when the pivot is the first element and better for random when it is the median. This leads the choice of pivot to perform favorable for one type of input at a time (either increasing, decreasing, or random). Bubble sort has worst-case $O(n^2)$, where n is the number of items being sorted, and best case O(n) when the input is already sorted. Bubble and insertion sort have the fastest best case guarantees, O(n). Bubble, quick, and insertion sort have the slowest worst case running times which is O(n^2).

When the input is in increasing order insertion, bubble and quick sort take substantially less time. Decreasing takes the longest for those three also . Random can represent an average performance since it is most represented of a typical input for a sort. Merge sort appears to be the fastest.