Generally speaking, insertion sort is used for smaller data sets. Larger data sets will be sorted with heap sort, merge sort, and quick sort. This makes sense given the worst, best, and average case scenarios for each method of sorting. Insertion sort is most useful for smaller data sets, since larger data sets would take too long, given that the worst and average case take exponentially longer as the number of elements increases. The other three methods, however, take comparatively similar lengths of time to sort, and the time taken is log-based. Our merge function doesn't work (since it didn't for 2a) so this sort was skipped.

Heapsort

- Worst case performance
  $O(n \log n)$
- Best case performance
  $O(n \log n)$
- Average case performance
  $O(n \log n)$

Insertion Sort

- Worst case performance
  $O(n^2)$
- Best case performance
  $O(n)$
- Average case performance
  $O(n^2)$

Merge Sort

- Worst case performance
  $O(n \log n)$
- Best case performance
  $O(n \log n)$
- Average case performance
  $O(n \log n)$

Quicksort

- Worst case performance
  $O(n^2)$
- Best case performance
  $O(n \log n)$
- Average case performance
  $O(n \log n)$