

# Breakdown of DataStructures

## Comparison of sorting algorithms on each, vector, linkedlist, and hashtable data structures.

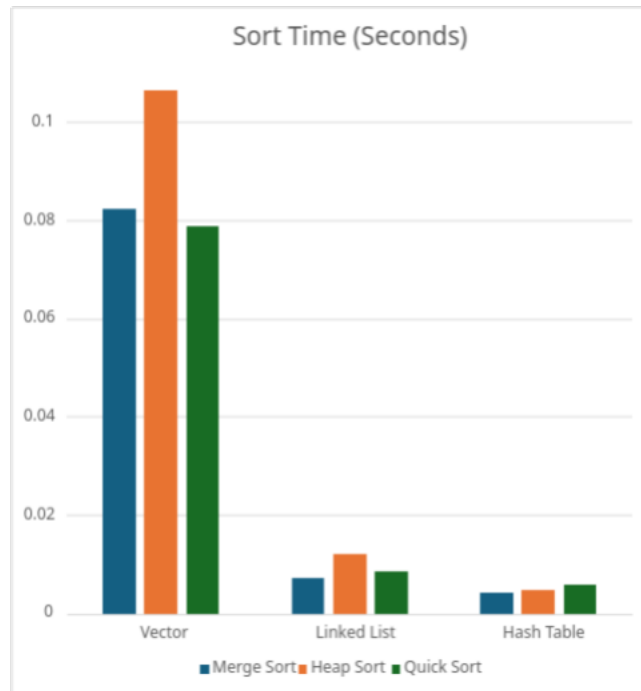
We will be comparing the following three sorting algorithms based on their completion in seconds:

- MergeSort
- HeapSort
- QuickSort

This experiment was conducted with the same data with a length of 12023 items. This data will first be loaded into the data structure before performing a sorting algorithm on the data. This process will be continued for each algorithm and data structure.

The data holds bid information, this data is originally stored in a .CSV file. its contains 20 different columns but for this program we will be sorting based on the auction id. The auction id is a 5 digit number that is unique.

## Graphical Overview



### 1. Vector

- Merge Sort: 0.0823 seconds
- Heap Sort: 0.1065 seconds
- Quick Sort: 0.0789 seconds

### 2. LinkedList

- Merge Sort: 0.0073 seconds
- Heap Sort: 0.012 seconds
- Quick Sort: 0.0086 seconds

### 3. HashTable

- Merge Sort: 0.00423 seconds
- Heap Sort: 0.0049 seconds
- Quick Sort: 0.006 seconds

## Summarization

- **Vector** From the data, we can confirm that the vector data structure had the slowest sorting times. The slowest sorting algorithm was the heap sort with an average time of 0.1065.
- **LinkedList** The linkedlist is significantly faster than the vector but still slower than the hashtable. The linked list is in most cases about 10 times quicker than the vector data structure.
- **HashTable** The hashtable consistently performed the sorting algorithms the quickest. The fastest sorting algorithm on average is the merge sort with an average time of 0.00423. On average the hash table was able to sort about twice as fast when compared to the linkedlist.