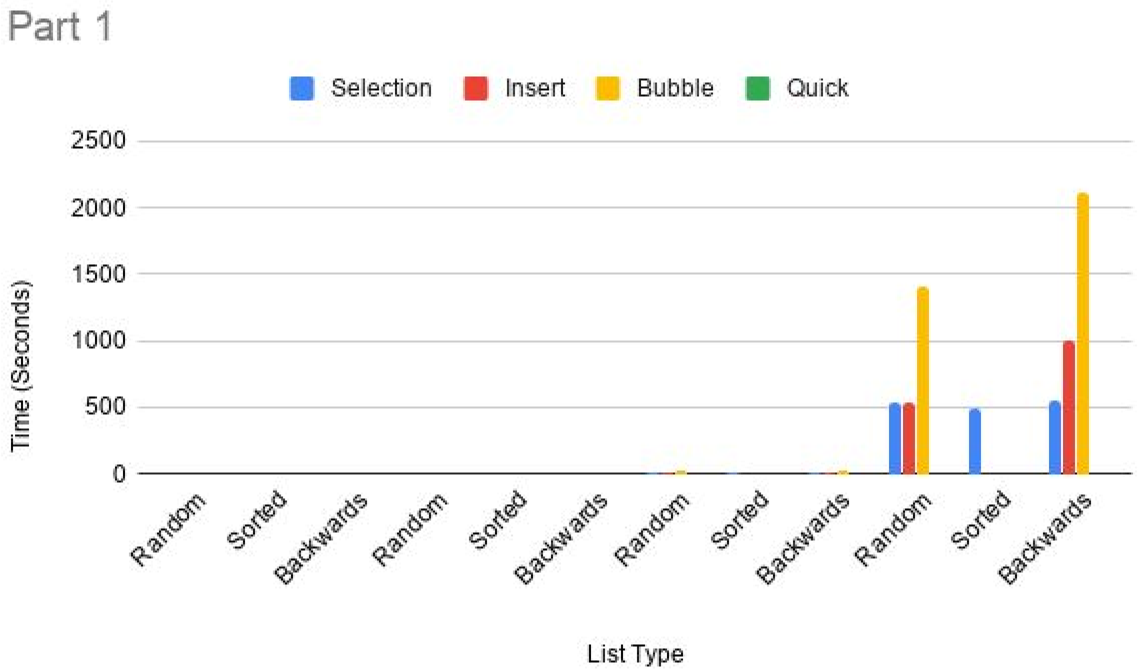
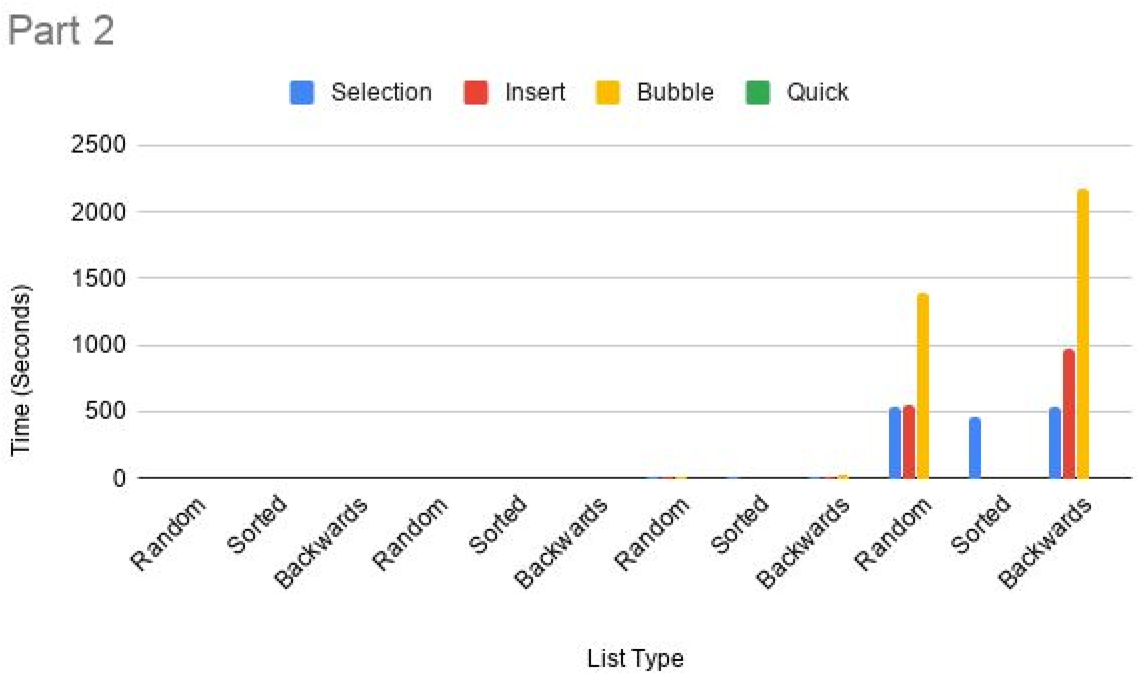


Part 1: c)
1,000 integers list:
Random List: Quick sort was much faster than any other algorithm. The slowest was bubble sort.
Sorted List: Insertion and Bubble sort were tied on time for the fastest algorithm. The slowest was quick sort by a hundredth of a second.
Backwards List: Quick sort was the fastest by tenths of a second. Bubble sort was the slowest.

At 10,000 integers lists, the quick sort began to output recursion errors because of how many sublists were being stacked on top of each other. Further data omits this algorithm.

100,000 integers list:
Random List: Selection Sort was the fastest by a few seconds compared to insertion sort. The slowest was bubble sort by over 1,000 seconds.
Sorted List: Bubble sort was the fastest by a few hundredths of a second compared to insertion sort. Selection was the slowest by 493 seconds.
Backwards List: Selection was by far the fastest sorting algorithm. Bubble sort was the slowest algorithm by over 1,100 seconds.



Part 2:
The modified function gave slightly longer runtimes and created a runtime error at the random 10,000 integer list compared to the starting at the sorted 10,000 integer list prior.

Part 3:
The radix function blew all other functions out of the way based on speed by drastically running faster in all list sizes and sorted types.