Sorting Evaluation Report (Resubmission)

Fun Facts:

- Merge sort for doubly linked lists now performs about 700x better than before
- Quick sort for doubly linked lists now performs about 300x better than before for the 6 digits case.

Surprisingly, we found that quicksort performs better than our merge sort for dll's even after the changes. Knowing that the time complexity of merge sort is always O(nlogn) and the time complexity of quick sort is on average the same but has a worst case of O(n^2), we naturally expected quick sort to be a little bit slower. However, we find that merge sort is slower which is likely due to memory allocation since we had to create new temporary nodes for each function call. Meanwhile, for our quick sort algorithm, we do not create any new temporary nodes and instead alter the dll in place. We believe these to be the main reasons why our quick sort is quicker than merge sort after our resubmission changes.

Overall, by implementing a true merge sort and not recalculating the tail and head in our quick sort function, we have significantly improved the performance of merge sort and quick sort for our doubly linked lists. In the screenshot below, you can see the runtimes for each sorting algorithm after two different executions of the program.

```
Algorithm           Number of Digits    DLL TIme (ms)       Vector Time (ms)
   ---------------------------------------------------------------------------
Quick Sort:         4                   0.023639            0.038822
Insertion Sort:     4                   0.071349            0.185518
Merge Sort:         4                   0.033039            0.272498
Quick Sort:         5                   0.28172             0.471769
Insertion Sort:     5                   9.86429             17.8638
Merge Sort:         5                   0.47496             2.79829
Quick Sort:         6                   3.53435             5.95247
Insertion Sort:     6                   2712.37             1764.84
Merge Sort:         6                   5.89737             32.257
-------------------------------------------------------------------------------
(base) sam@sam-ZenBook-UX325UA:~/Documents/Project-1-Sorting-Evaluation-CPSC-223$ ./main
Algorithm           Number of Digits    DLL TIme (ms)       Vector Time (ms)
   ---------------------------------------------------------------------------
Quick Sort:         4                   0.025002            0.041105
Insertion Sort:     4                   0.068613            0.185766
Merge Sort:         4                   0.033089            0.267163
Quick Sort:         5                   0.282738            0.470517
Insertion Sort:     5                   9.80877             17.5534
Merge Sort:         5                   0.480815            2.73804
Quick Sort:         6                   3.52446             5.91129
Insertion Sort:     6                   2706.75             1764.62
Merge Sort:         6                   6.08034             31.0357
-------------------------------------------------------------------------------
```