The program compares different sorting mechanisms for integers in a linked list. The different methods for sorting the integers in the linked list were bubble sort, merge sort, and quicksort. The goal is to sort the linked list and get the median value in length n linked list using known methods of a linked list class.

Bubble sorting the linked list is the method that uses the less amount of lines of code. In my opinion it is also the least complex to sort data. The big O notation for bubble sort is O (n^2) due to its two while loops that depend on a Boolean expression and the linked list of size n being traversed. Using a linked list of size 5, it managed to make 8 comparisons in total. Merge sort has a big O notation of O (n log n). It does use several recursion calls but managed to be the fastest with the least number of comparisons in a linked list of size 5 with only 7 comparisons. Quick sort also has a big O notation of O (n log n) but makes the most comparisons at 16 with a linked list of size 5. Due to their big O notation, bubble sort may be close to speed in terms of comparisons made with merge sort but due to having n^2 as its complexity, the higher the n, the more comparisons it will make. This is proven when the linked list is of size 25. Bubble sort makes 408 comparisons, merge sort makes 87, and quick sort makes 212 making bubble sort the least effective. The following table shows that when n = 5 there isn’t much difference in complexity but when n = 25 there is a significant difference.

What I learned from this project is that even though a sorting method may be easier, or less complex, to type, it isn’t necessary the most effective in complexity. It is also important to know the size of the data that will be sorted because it may not matter the complexity if the difference, in this case comparisons, is insignificant.

Appendix:

Line 9 to 12 Class Node

Line 14 to 17 Class List

Line 20 to 27 Append

Line 30 to 36 Copy

Line 39 to 40 IsEmpty

Line 43 to 49 GetLength

Line 52 to 65 BubbleSort

Line 68 to 85 MergeSort

Line 88 to 110 Merge

Line 113 to 131 QuickSort

Line 134 to 145 QuickSortMerge

Line 148 to 153 ElementAt

Line 156 to 160 Median

Line 163 to 168 Print

Line 170 to 207 Creates linked list and adds integers to nodes; prints the median

Line 210 to 230 Tests sorting methods by comparisons