Insertion Sort

Sort & Input	Unsorted Linked List	Sorted Linked List	Reverse Sorted Linked List
10	230,300 ns	71,700 ns	196,800 ns
50	3,869,400 ns	379,000 ns	3,541,200 ns
100	3,205,400 ns	106,600 ns	4,505,500 ns
150	3,740,300 ns	104,500 ns	5,840,000 ns
300	17,760,400 ns	253,800 ns	35,434,800 ns
500	69,537,300 ns	466,300 ns	174,194,800 ns
1,000	782,471,800 ns	1,472,400 ns	919,381,100 ns
1,500	1,508,913,700 ns	3,895,400 ns	2,348,695,200 ns

Quick Sort

Sort & Input	Unsorted Linked List	Sorted Linked List	Reverse Sorted Linked List
10	33,600 ns	28,000 ns	26,400 ns
50	281,500 ns	184,400 ns	214,500 ns
100	1,107,600 ns	332,100 ns	298,900 ns
150	636,400 ns	515,600 ns	552,400 ns
300	2,018,500 ns	1,368,200 ns	1,140,800 ns
500	4,676,800 ns	3,540,500 ns	3,855,900 ns
1,000	17,032,700 ns	13,436,500 ns	17,200,100 ns
1,500	51,243,900 ns	42,853,100 ns	48,121,600 ns

Merge Sort

Sort & Input	Unsorted Linked List	Sorted Linked List	Reverse Sorted Linked List
10	51,200 ns	34,000 ns	50,200 ns
50	205,500 ns	358,000 ns	151,600 ns
100	506,100 ns	409,300 ns	260,500 ns
150	625,700 ns	479,400 ns	423,100 ns
300	1,212,700 ns	701,800 ns	752,200 ns
500	1,879,700 ns	1,551,000 ns	1,515,000 ns
1,000	4,391,800 ns	4,267,500 ns	3,480,700 ns
1,500	8,717,100 ns	6,691,100 ns	6,541,300 ns

Insertion Sort

The while loop within the method runs to O(n). And the recursive portion of it runs also O(n). Thus the Average and Worst Case runs to O(n^2). In the Best Case, its already sorted, and the while loop doesn't run so it would be O(n)

Quick Sort

The while loop within the method runs to O(n). The recursive portion of it runs O(log n) because of the divide and conquer method. The Average and Best Case runs to O(nlog n) while the Worst case runs to O(n^2) if the pivot happens to be the greatest item. For this reason, the Reverse Sorted Linked List and Sorted Linked List runs at Best time because my algorithm chooses the pivot in the middle index, which will always choose the middle item in size, it splits the list in half exactly meaning the Average/Best case time O(nlog n). (Note: If my pivot was taken at the last index, then the Reverse Sorted would be the Worst Case)

Merge Sort

The while loop within the merging portion runs to O(n). The recursive portion of it runs O(log n) because of the divide and conquer method. All Cases run to O(nlog n).