SOFT153 Coursework

1. Doubly linked list:
2. Please find attached a Visual Studio solution with the linked list functionality.
3. Please find attached a Visual Studio solution with the linked list sorting functionality.
4. Run-time complexity

Quicksort

|  |  |
| --- | --- |
| Nodes | Time (ms) |
| 2000 | 34.9992 |
| 4000 | 148.9959 |
| 8000 | 653.9827 |
| 16000 | 3708.9952 |
| 32000 | 21066.443 |
| 64000 | 93996.6672 |
| 128000 | 465144.8786 |

Insertion Sort

|  |  |
| --- | --- |
| Nodes | Time (ms) |
| 2000 | 13.9997 |
| 4000 | 23.999 |
| 8000 | 99.9974 |
| 16000 | 441.9883 |
| 32000 | 2229.9427 |
| 64000 | 9776.741 |
| 128000 | 43889.2841 |

Comparison

As you can see when both insertion sort and quicksort are plotted on the same graph, insertion follows the O(n) time complexity pattern and quicksort clearly shows it more closely represents O(n log n). After some research it appears that my insertion sort is seemingly more efficient than the average insertion sort which is O(n2).