Suzanne Townsend

Student Id: P2996646

Date: 2017/09/13

PROGRAMMING III

AT1.3 Bubble Sort

TABLE OF FIGURES

**Pseudo Code:**

[i Psudo Code 1](#_Toc493079929)

**Computational Complexity:**

[i Computational Complexity, Benefits and Pitfalls. 1](#_Toc493079930)

**Test Table:**

[i Test Table 2](#_Toc493079980)

**Reference to code:**

[i Reference to code 3](#_Toc493080019)

**Testing Evidence:**

[i Lowercase - Not Sorted. 3](#_Toc493080046)

[ii Lowercase - Sorted 3](#_Toc493080047)

[iii Numbers - Not Sorted. 4](#_Toc493080048)

[iv Numbers - Sorted. 4](#_Toc493080049)

[v Lower and Uppercase - Not Sorted. 4](#_Toc493080050)

[vi Lower and Uppercase - Sorted. 5](#_Toc493080051)

**References:**

[i References 5](#_Toc493080536)

**PSEUDO CODE:**

|  |
| --- |
| Public void BubbleSort (linkedList<> m)  Outter loop for ( j =0; j = m.Size; j++)  Inner loop for ( i = 0; i = m.Size; i++)  If m.( j ) > j + 1  Swap [ j ], list[ j + 1 ]  End if.  End for .  End for .  End Void. |

i Pseudo Code.

**BENEFITS, PITFALLS AND THE COMPUTATIONAL COMPLEXITY OF THE ALGORITHM:**

|  |
| --- |
| For this program, I have chosen Bubble sort. Bubble Sort has a Big-O Complexity of O(n2), Where n is the number of items being sorted as depicted in the chart it forms a linear climb as the number of elements increase. There for while bubble sort is almost instantaneous for smaller programs dealing with lesser amounts of data as the elements increase the sort becomes slower than that of comparison to a Merge sort. |

i Computational Complexity, Benefits and Pitfalls.

**TEST TABLE:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Is it errorTrapped.  i.e null entry. | Does it add to the listBox | Does it display the node number | Does it do its intended job ie. Add before | Can the program be crashed. | Is it user friendly. |  |
| addFirst | Yes | Yes | Yes | Yes | No | Yes |  |
| addLast | Yes | Yes | Yes | Yes | No | Yes |  |
| Find | Yes | Yes | N/A | Yes | No | Yes |  |
| addAfter | Yes | Yes | N/A | Yes | No | Yes |  |
| AddBefore | Yes | Yes | N/A | Yes | No | Yes |  |
| Delete | Yes | Yes | N/A | Yes | No | Yes |  |
| Clear | N/A | N/A | N/A | Yes | No | Yes |  |
| Sort | Yes | Yes | Yes | Yes | No | Yes |  |

i Test Table

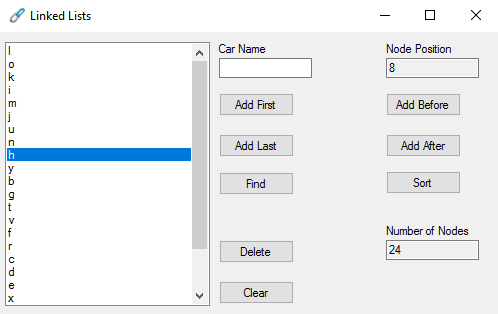
**REFERENCE TO CODE:**

A screenshot of a cell phone

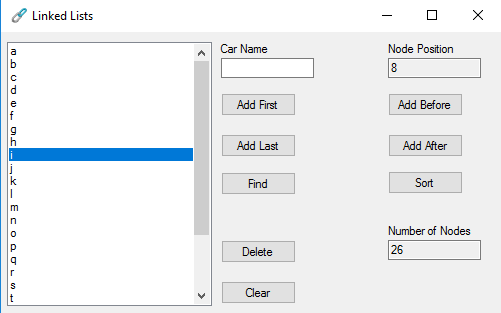
Description generated with very high confidence

i Reference to code

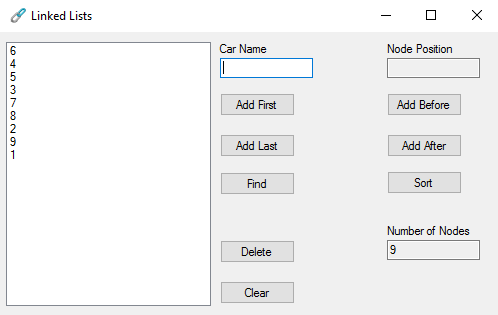
**TESTING EVIDENCE:**



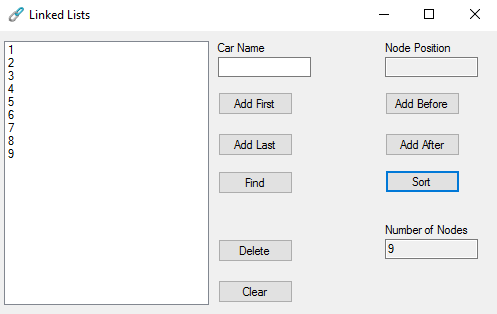
i Lowercase - Not Sorted.



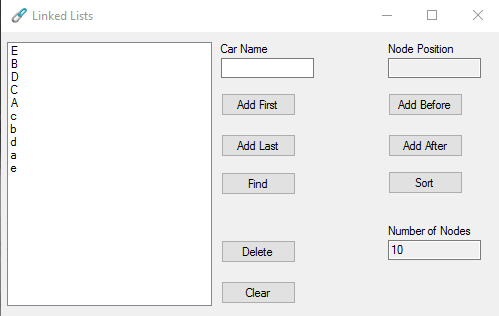
ii Lowercase - Sorted



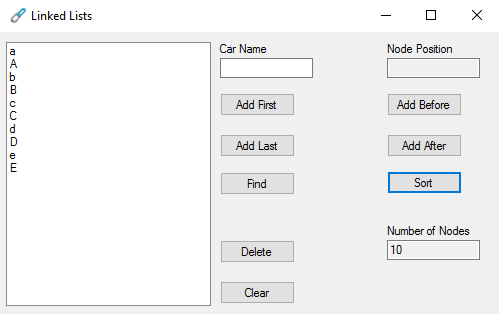
iii Numbers - Not Sorted.



iv Numbers - Sorted.



v Lower and Uppercase - Not Sorted.



vi Lower and Uppercase - Sorted.

**REFERENCES:**

|  |
| --- |
| <https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm>  https://ecampus.polytechnic.wa.edu.au |

**i** References