Sorting Algorithms API Documentation

# Overview

In this project a Custom Linked List Implemented in C++. It is used to store **n** number of items in the List. The Item can be added, deleted and can get head of the Linked List.

* Linked List has head which is the starting of the list.
* Items can be added at the start of the Linked List.
* Items can be added at the end of the Linked List.
* Items can be deleted from the start of the Linked List.
* Items can be deleted from the end of the Linked List.
* Start of the List can be extracted.

# Methods and Functions of Linked List

# Add At Front:

Item can be added at the front the of the Linked list. If there is a linked list having items

4 -> 7 -> 2 ->8. When we add element 9 to this list it will added at 9 -> 4 -> 7 -> 2 -> 8

**Example:**

List.addNodeAtFront(node);

# Add At End

Item can be added at the End of the Linked list. If there is a linked list having items

4 -> 7 -> 2 ->8. When we add element 9 to this list it will added at 4 -> 7 -> 2 -> 8 -> 9

**Example:**

List.addNodeAtEnd(node);

# Delete At Front:

Item can be deleted at the front the of the Linked list. If there is a linked list having items

4 -> 7 -> 2 ->8. When we delete element from list it will deleted and the list will be 7 -> 2 -> 8

**Example:**

List.deleteNodeAtFront(node);

# Delete At End

Item can be deleted at the end the of the Linked list. If there is a linked list having items

4 -> 7 -> 2 ->8. When we delete element from list it will deleted and the list will be 4 -> 7 -> 2

**Example:**

List.deleteNodeAtEnd(node);

# Get Head

The head of the Linked List can be retrieved. If the list is the 4->8->6->2 then gethead function will return 4.

**Example:**

List.getHead();

# Methods and Functions of Sorting Algorithms

# Selection Sort:

Item can be added at the front the of the Linked list. After that the algorithm compares the values of each item and then sort them accordingly.

**Example:**

List.SortBySelectionSort();

# Insertion Sort

Item can be added at the front the of the Linked list. After that the takes each node and then sort them according to their place in the linked list.

**Example:**

List.SortByInsertionSort();

# Built-in Sort

Item can be added at the front the of the Built-in Linked List. After that the Built-in sorting Algorithm takes each node and then sort them according to their place in the linked list.

**Example:**

List.sort();