Assignment 2: Sorted List ReadME

Abhishek Prajapati
Eric Cajuste

Implementation

This assignment uses a singly linked list to implement sorted linked list. It reads in data from main method like integers, double, strings, and characters. It then creates a sorted list and inserts the data in defending order. Also it does not allow duplicate data entry. The program works as following, when some inputs are give.
INPUT: 9, 5, 4, 100, 100, 999
OUTPUT: 999, 100, 9, 5, 4
remove 9 : 999, 100, 5, 4

Analysis

SLCreate

This Function allocates a constant amount of space of space for a SortedList struct. This method runs in O(1) time because it only allocates memory for the sorted list ones.

SLCreateIterator

This function allocates a constant amount of space for a SortedListIterator struct. This function runs in O(1) time because it only allocates memory one time as well.

SLInsert

This function takes in a pointer to the object to be inserted in the sorted list. This method will iterate through the sorted list and insert the data into its appropriate location keeping the sorted list sorted in descending order. This function runs in O(n) time because there is a chance we have to iterate through end of the list to insert a new object. Therefor it has to compare n elements.

SLRemove

This function takes in a pointer to the object to be removed from the sorted list. It iterates through the sorted list and find the object to be removed and removes the object from sorted list. This function runs in O(n) time because there is a chance we have to iterate through end of the list before we find out object that we want to remove.

SLDestroy

This function destroys the sorted list and frees it from the memory. This function runs in O(n) times because we have to remove n elements from the sorted list.

SLNextItem

This function returns the next item from the sorted list. This function runs in O(1) time because it only returns one pointer object at a time.

SLGetItem

This function returns current object that is being pointed by the SortedListPointer iterator. This function runs in O(1) time because it only returns the data from the current pointer.

SLDestroyIterator

This function frees the pointer to an iterator. The function runs in O(n) time because we have to delete m nodes the iterate is pointing to.

NOTE

Floating point numbers prints up to six digits max.
i.e) 1.1234567 will print 1.123456
also 1.123 will print 1.123000. It will add zeros in the end up to 6 digit max after the '.'