

## Sorted List Project Documentation

Jimmy Le<sup>1</sup>, Carlos Dominguez<sup>2</sup>

Rutgers University

<sup>1</sup>jl1415@rutgers.edu

<sup>2</sup>cmd363@rutgers.edu

## Sorted List

Our Sorted List library is a robust solution for storing data in easy to manage linked lists.

## Functions

### **SortedListPtr SLCreate(CompareFuncT cf, DestructFuncT df)**

The caller specifies a comparator function that should take in two data inputs and returns -1 if the first object is smaller, 0 if the two objects are equal, and 1 if the first object is larger. The caller also specifies a destructor function that properly frees the memory the object was allocated. SLCreate will return a pointer to the head of a linked list, and the caller can then specify a data value for that node.  $O(1)$ .

### **SLDestroy SortedListPtr sl**

Destroys the linked list given a pointer to the head of the node.  $O(n)$ ,  $n$  being the size of the sorted list.

### **int SLInsert(SortedListPtr sl, void\* o)**

Inserts the object 'o' into the linked list passed in the function, in its proper place according to list order. If the object is added into the list successfully, SLInsert returns a 1. Otherwise, it returns a 0 (such as if the object already exists in the list).  $O(n)$ ,  $n$  being the size of the sorted list, and if the object is being inserted into the end of the list.

### **SortedListIteratorPtr SLCreateIterator(SortedListPtr sl)**

Creates an iterator over the given linked list that can move down the list and retrieve data from each node.  $O(1)$ .

### **void SLDestroyIterator(SortedListIteratorPtr iter)**

Destroys the given iterator.  $O(1)$ .

### **void\* SLNextItem(SortedListIteratorPtr iter)**

Returns the next piece of data encapsulated by the iterator and moves the iterator down the list.  $O(1)$ .

## Using Our 'main.c'

Configure the driver for different types of input. By default, it is meant for taking integer inputs and then sorting it. Run the driver, then enter one input and press enter. Repeat until all the inputs have been entered, then enter a breakpoint input (by default, any letter). The main.c will then output the sorted list. The driver does have an issue comparing strings—a different main.c would be required for doing so.