

lab 12 Basic sorting: Selection and Insertion Sort

Part 1: Implement Selection and Insertion sort for your Array List class. Then implement any of the basic sorts (Bubble, Selection, or Insertion) for a singly Linked List. **Note:** Keep your old implementation and append the following functions:

```
1
2 template <class T>
3 class Array {
4     private:
5         /* You fill out the private contents. */
6
7     public:
8         /* Runs a bubble sort algorithm on the array.
9          * The array shall be ordered from least to greatest
10          */
11         void bubbleSort();
12
13         /* Runs a selection sort algorithm on the array.
14          * The array shall be ordered from least to greatest
15          */
16         void selectionSort();
17
18         /* Runs a insertion sort algorithm on the array.
19          * The array shall be ordered from least to greatest
20          */
21         void insertionSort();
22
23         /* Runs the sort routing you believe is the best. */
24         void sort();
25
26 };
27
28 /* SLL = Singly Linked List */
29 template<class T>
30 class SLList {
31     public:
32         /* Sort the linked list. You may use any sort algorithm you wish */
33         void sort();
34
35 };
```

Write some test cases:

Create some test cases, using cxxtestgen, that you believe would cover all aspects of your code.

Part 2: Performance

Generate a graph to compare the performance of bubble sort, selection sort, insertion sort, and the

sort you chose for a Singly Linked List. Your graph should have data size on the x axis and time on the y axis. Make sure to label each graph line! Please turn in as a .pdf!

Auto Grader:

The auto grader is only grading part 1, I will have to assess part 2. In other words, if the auto grade issues a 100, that is only for part 1!

Memory Management:

Now that are using new, we must ensure that there is a corresponding delete to free the memory. Ensure there are no memory leaks in your code! Please run Valgrind on your tests to ensure no memory leaks!

How to turn in:

Turn in via GitHub. Ensure the file(s) are in your directory and then:

- `$ git add <files>`
- `$ git commit`
- `$ git push`

Due Date: February 22, 2017 2359

Teamwork: No teamwork, your work must be your own.