CSC 212: Data Structures and Abstractions
Spring 2018
University of Rhode Island
Weekly Problem Set #9

Due Thursday 4/5 before class. Please turn in neat, and organized, answers hand-written on standard-sized paper **without any fringe**. At the top of each sheet you hand in, please write your name, and ID. The only library you're allowed to use in your answers is `iostream`.

1 Quick Sort

1. Implement Quick Sort.
2. Describe how quick sort performs when all the elements of the list are: a) The same. (EX: [1, 1, 1, 1, 1]) b) Sorted order. c) Reverse order. d) Random order.

2 Linked Lists

1. Using this weeks' lab as a basis, create a new method `void reverse()` that reverses the linked list.
2. For each of the following please provide the time complexity for a Singly Linked List (SLL), as well as a Doubly Linked List (DLL).

Function	SLL	SLL without Tail	DLL
`int size();`			
`int at(int);`			
`int front();`			
`int back();`			
`bool empty();`			
`void clear();`			
`void set(int, int);`			
`void push_back(int);`			
`int pop_back();`			
`void insert(int, int);`			
`void erase(int);`			
`void reverse(int);`			

3. Implement both `insert_at(int index)` as well as `remove_at(int index)` for Doubly Linked Lists.

The following problems are considered optional:

1. Compare your partition scheme to the Lomuto Partition Scheme. Why does the partition scheme of quick sort matter? Can any partition scheme reduce time complexity below the worst case of $O(n^2)$?