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.
3. 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 $O(n^2)$?

2 Linked Lists

1. Think of two use cases for linked lists, justify your response. (Hint: Think about the circumstances where linked lists perform the best.)
2. Using this weeks' lab as a basis, create a new method `void reverse()` that reverses the linked list.
3. Explain the time-complexity differences that would occur if the linked list class you built didn't have a `tail` element. What operations would suffer? Would any improve?
4. For each of the following please provide the time complexity for a singly linked list, as well as a doubly linked list.

Function	Singly Linked List	Doubly Linked List
`int size();`		
`int capacity();`		
`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);`		