CSC 212: Data Structures and Abstractions

Fall 2018

University of Rhode Island

Weekly Problem Set #7

Due Thursday 11/8 at the beginning of 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.

1. For each of the following please provide the worst case time complexity for a Singly Linked List (SLL), as well as a Doubly Linked List (DLL).

Function	SLL	SLL without Tail	DLL
`int size();`	n	n	n
`int at(int);`	n	n	n
`int front();`	1	1	1
`int back();`	1	n	1
`bool empty();`	1	1	1
`void clear();`	n	n	n
`void set(int, int);`	n	n	n
`void push_back(int);`	1	n	1
`int pop_back();`	n	n	1
`void insert(int, int);`	n	n	n
`void erase(int);`	n	n	n
`void reverse();`	n	n	n

2. How many pointers are required for a SSL of length n? How many pointers are required for a DLL of length n?
   n + 1; 2n + 2

3. Describe how to reverse a SLL without a tail.
   Create 3 nodes, curNode, prevNode and nextNode. Initialize them as curNode = head and prevNode = null. Use curNode to iterate over the list. Set the nextNode to curNode.next and set curNode.next to prevNode. Set prevNode to curNode and curNode to nextNode. When curNode is null, set the head to prevNode.

4. What is required to change a linked list into a queue? What about a stack?
   Modifying the API to allow for fewer options, usually just push, pop, and sometimes peek.
   In specific, a queue will only have append and pop_back (as push and pop), no remove_at, and will require a head and tail node. On the other hand, a stack will have a push_back and pop_back (as push and pop), no remove_at, and only a head node (typically as "top"). If a peek function is used, it would be the same as the front function.

5. Draw the state of an empty stack and an empty queue after running the following functions for each:

   ```
   push(1); push(2); push(3); pop(); push(4); pop(); pop(); push(5)
   ```

   Stack: [1, 5]; Queue: [4, 5]