CMSC 341 Data Structures
List Review Questions

October 26, 2010

Please refer to the textbook for List, ListNode, and List::iterator class definitions. These definitions are the same as those found in the class notes. You may assume that all member functions of these classes have been written and work properly when answering the questions below.

1. Write a new member function of the List class named **ReversePrint** that uses iterator to display the elements of the List in reverse order. The one and only parameter to ReversePrint is the output stream to which the elements are displayed. The data elements should be enclosed in angle brackets ("< >") and separated by commas. Do not construct a copy of the list that is in reverse order, just use iterators. The prototype for **ReversePrint( )** is shown below

   ```
   void ReversePrint( );
   ```

2. Write a new function named **Splice( )** whose prototype is shown below. ***Note that Splice( ) is not a member function of the List class***. This function "splices" L2 into L1 at the specified position (pos is an iterator over L1). If pos is past end, **Splice( )** does nothing. For example suppose L1 = {1, 2, 3, 4, 5 } and L2 = {10, 20, 30 } and pos is constructed as list1.first( ) and then advanced twice so it is positioned at the "3". Then the function call L1.Splice(L2, pos); causes L1 to become { 1, 2, 3, 10, 20, 30, 4, 5 } and L2 is unchanged.

   What is the asymptotic performance of **Splice( )**? Bear in mind that there are two lists which may be of different lengths.

   ```
   public static
   void Splice( List<AnyType> L1, const List<AnyType> L2, Iterator<AnyType> pos);
   ```

3. Complete the following table of Big-Oh, worst-case asymptotic time performance for the given operations and implementations of the List ADT. Give your answers in terms of, n, the number of elements in the list.

Operation	Double Linked List	ArrayList
insert		
find		
remove		
makeEmpty		
push_back		
front		

4. Suppose you are provide with a set of N random numbers which are to be inserted into a sorted List (smallest to largest). What would be the worst-case asymptotic time performance for building the entire list?

5. One advantage of using a double-linked list is the availability of bi-directional iterators – iterators that move "forward" via operator++ and move "backward" via operator–. Suppose that in order to save memory, you (or your boss) decide to use a single linked list. Can a single linked list still support bi-directional iterators? If so, briefly describe how the iterator would move forward and backward. Be sure to include the Big-Oh performance of these operations. If bi-directional iterators are not possible with a single linked list, explain why not.

6. Describe the advantages (if any) and disadvantages (if any) of each of the List ADT implementation – singly linked list, circular linked list, doubly linked list, doubly circular linked list, arraylist, cursor space.