Project 2: Implementation of the Merge List Function (for Merger of Unique Elements) for Singly Linked List and the Time Complexity Analysis

Due by: Feb. 27th, 11.59 PM

In the code given to you for this project, you can notice in the main function that you will be creating two lists (called firstIntegerList and secondIntegerList) of the same size (listSize) and whose values are in the range [1...maxValue]. Add a member function *mergeList* to the Singly Linked List-based implementation of the List ADT. The member function takes as input a List object (representing the second integer list) as parameter and appends it to the List object (representing the first integer list) on which the function will be called. You should append only those elements (from the second integer list) that are not already in the first integer list. Feel free to create additional member functions to facilitate this.

For example: if the firstIntegerList is 11 --> 22 --> 44 --> 55 --> 78 --> 89 and the secondIntegerList is 22 --> 56 --> 89 --> 77 --> 32 --> 55, then the contents of the firstIntegerList (after the merger) should be : 11 --> 22 --> 44 --> 55 --> 78 --> 89 --> 56 --> 77 --> 32.

The main function has the timers setup to merge the two lists and print the merging time in milliseconds. You should run the code for the following combinations of values for the two variables and tabulate the results.

maxValue: 1000, 10000, 100000
listSize: 1000, 10000, 100000

Plot bar charts (in Excel) of the logarithm values of the run time (since the run time values will be of different ranges for the above combinations of values, we will take the natural logarithm, to the base e, in Excel and use these values to plot) as follows:
(1) maxValue in X-axis and the listSize in Y-axis
(2) listSize in X-axis and the maxValue in Y-axis

Submission (in Canvas):
Submit a singly report that has the following:
(a - 20 pts) Explain the logic/algorithm that you incorporated to your code to accomplish the merger of only unique elements: i.e., to merge an element (from the secondIntegerList) to the firstIntegerList only if the element is not already in the latter. Provide a pseudo code for the entire merge list function (including the logic for merger of unique elements) and analyze its time complexity as a function of the list size, n.
(b - 45 pts) Include the complete C++ code comprising of the Node class, List class with all the added member functions and the main function.
(c - 11 pts) Tabulate the actual values of the run time (in milliseconds) for the combinations of maxValue and listSize given.
(d - 12 pts) Present Excel bar charts (using the two ways prescribed) for the run time (plotted in logarithm of the milliseconds observed)
(e - 12 pts) From the bar charts, infer whether (on a relative basis) there is a significant increase in the run time with increase in listSize (for a given maxValue) or there is a significant increase in the run time with increase in maxValue (for a given listSize)?