

CSCI 33500 - Spring 2016 - Homework #2, Covering chapters 3 and 4.

Due in class Monday March 7th

For this assignment, follow these pseudocode guidelines:

- all type declarations are considered implicit (but no penalty for including include them).
- brackets should still be present to define the scope of procedures/loops.
- use indentation to make these scopes easy to understand.
- make sure that * pointers are used correctly and when needed.
- be explicit and add comments liberally.

All questions are worth the same amount of points.

1 Chapter 3 - List, Stack, and Queues

- 1.1 Given a *simple* linked list L of size N, write a procedure `reverse(L)` that returns the list L in reverse order, using only the basic list operations. Give an analysis of the runtime of your procedure.

(The basic list operations are defined in chapter 3.3 / page 81-83, and are `size()`, `clear()`, `empty()`, `front()`, `push_front(x)`, `pop_front()`, `begin()` and the associated iterator methods. Note that `push_back(x)`, `pop_back()`, `end()` are not available for a simple linked list.)

- 1.2 Given two *sorted* simple linked lists, L1 and L2, of size N and M respectively, write a procedure to compute L1 XOR L2, the list of elements that are in either L1 or L2 but not in both, using only the basic list operations. Give an analysis of the runtime of your procedure.

- 1.3 Transform the following *infix* expressions to their *postfix* form (see section 3.6.3 / page 105 of the book, PEMDAS order of operation):

$a * x ^ 3 + b * x ^ 2 + c * x + d$

$d + x * (c + x * (b + x * a))$

For one of these transformations (of your choice), draw the stack and output as it changes for each stack operation. Follow the example given in page 109 of the book.

2 Chapter 4 - Trees

- 2.1 Draw the binary trees corresponding to the expressions of question 1.3. Parentheses should not appear.
- 2.2 Prove by *induction* that in a tree of N nodes, there are $N+1$ `nullptr` links representing children.
- 2.3 Show the steps and final result of inserting 5, 1, 3, 8, 10, 9 (in this order) into an initially empty AVL tree. At each step, show the tree before and after balancing, and state if you are performing a simple/double rotation with the left/right child.

3 Extra Credit

Write recursive versions of the procedures you've written for 1.1 and 1.2.