CMPS 1500 Introduction to Computer Science I – Fall 13

10/21/13

7. Homework

Programming portion due **10/29/13** at 11:55pm on Blackboard.
Written portions (parts (b) and (c) of each question) due **10/30/13** at the beginning of class.

Please create a single Python file for the homework, and use the following naming convention: `lastName_firstName_hw7.py`. The written portions of each question can be turned in on paper.

In order to receive any credit for the programming portions, you are required to thoroughly comment and test your code.

1. **Find i-th element (7 points)**
   (a) (4 points) Write a function that takes as input the front node of a linked list, and a non-negative number `i`. The function should return the data stored in the `i`-th node of the list. Just as in an array, the numbering of the node should start at 0. Test your code.
   (b) (1 points) What is the asymptotic running time of your function in terms of `n`, where `n` denotes the length of the input list?
   (c) (2 points) Consider running your code on the following list: `L: 2`→ `12`→ `22`→ `32`→ `42`→ with `i=3`. Trace your code by showing, on paper, the changes in the variables and in memory during the execution of your function.

2. **Length of list (7 points)**
   (a) (4 points) Write a **recursive** function that takes as input the front node of a linked list and returns the number of nodes in the list. Test your code.
   (b) (1 points) What is the asymptotic running time of your function in terms of `n`, where `n` denotes the length of the input list?
   (c) (2 points) Consider running your code on the following list: `L: 'a'`→ `'b'`→ `'c'`→. Trace your code by showing, on paper, the changes in the variables and in memory during the execution of your function.

3. **Linking two linked lists (7 points)**
   (a) (4 points) Write a function that takes as input the front nodes of two linked lists, and links those two linked lists together. The second list should be linked to the end of the first list. There should not be any new nodes created. The function should return the front node of the result list. Test your function with several inputs.
   (b) (1 points) What is the asymptotic running time of your function in terms of `n`, where `n` denotes the total length of both input lists?
   (c) (2 points) Consider running your code on the following two lists: `L1:'a'`→ `'b'`→ `'c'`→ and `L2:'d'`→ `'e'`→ `'f'`→. Trace your code by showing, on paper, the changes in the variables and in memory during the execution of your function.