COMPUTER SCIENCE 21A (SPRING, 2022)
DATA STRUCTURES AND ALGORITHMS

PROBLEM SET 3

- For this assignment you can choose to use either Cormen-lib or the pseudocode we use in class.
- Your assignment should be submitted via Latte as a zip file containing either ps3.py and ps3_test.py or a text file (use the provided template).
- Zip both files into a zip named FirstnameLastname-PS3.zip
- Late submissions will not receive credit.

You only need to return a value if a problem specifies a return type.

Q1. Write a recursive function `sort_stack` that sorts a given stack in ascending order. Equivalent elements of the stack must retain their original ordering. What is the running time of your algorithm?

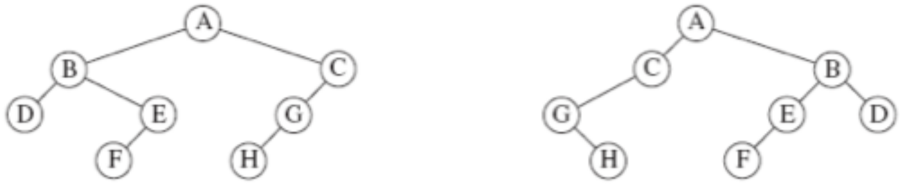
Q2. Write a recursive function `linked_list_min` that finds the minimum element in a singly linked list. What is the running time of your algorithm?

Q3. Write a function `weighted_sum` that returns the sum of the values stored in a binary tree of integers weighted by the depth of each value. You should return the value at the root plus 2 times the values stored at the next level of the tree plus 3 times the values stored at the next level of the tree and so on. What is the running time of your algorithm?

Q4. Define the lowest common ancestor (LCA) between two Nary tree nodes *v* and *w* as the lowest node in the tree that has both *v* and *w* as descendants (where we allow a node to be a descendent of itself). Given two Narry tree nodes *v* and *w*, write an efficient function `lowest_common_ancestor` that finds the LCA of *v* and *w*. What is the running time of your algorithm?

Q5. Write a non-recursive function `non_recursive_in_order` that, given the root of a binary tree, and the number of elements in the tree, preforms an in-order traversal of the binary tree, storing the elements in an array and returns it. Your function should run in linear time.

Q6. Two trees, T_1 and T_2, are isomorphic if T_1 can be transformed into T_2 by swapping left and right children of (some of the) nodes in T_1. For instance, the two trees below are isomorphic because they are the same if the children of A, B, and G, but not the other nodes, are swapped. Write a polynomial time function `is_isomorphic` that decides if the two trees are isomorphic.



<u>Practice problems not for grading</u>:

1. Solve the following recurrences, which in all cases have $T(0) = T(1) = 1$.

$T(n) = 4T(n/2) + n^2$

$T(n) = T(n/2) + \log n$

$T(n) = T(n/2) + n$