CSDS 233 Due October 02, 11:59 PM

Assignment 2: Binary Trees, Stacks, and Queues

Programming Assignment: Reverse Linked List (50 pts.)

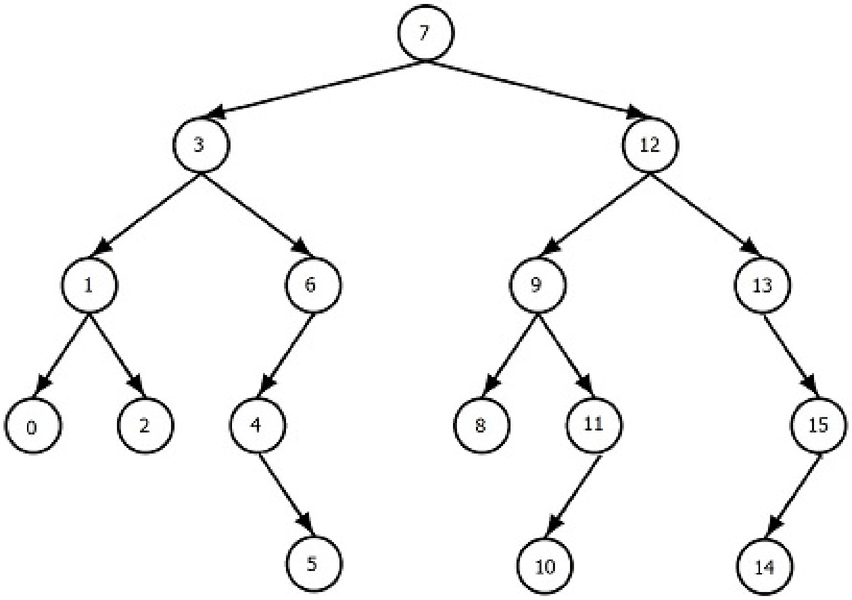
Create a method for reversing a singly-linked list *in place* without using another list. More formally, you want to be able to establish an upper bound on the number of variables created that is a single constant value, not a function of the size of the linked list or any other property of it. Include a Main method that generates a linked list of random integers (between, say, 1 and 100) and prints it before and after reversal.

The project will be graded based on the following:

- **20 pts**: Does the list emerge in reverse order?
- **20 pts**: Does the program run in constant space?
- **3 pts**: Are variable names descriptive and convey their purpose? Of course, simple concepts like a loop variable do not require descriptive names; e.g., it is perfectly fine, even preferable, to use i for a loop variable. Is formatting clear and consistent?
- **3 pts**: Comments should aid the reader to understand the code. Comments that restate what is already clear from the code are redundant and not helpful. Nor are comments that are not consistent with the code. For each method implementation, state in the comments its worst-case running time.
- **4 pts**: Are there lines that never execute? Inelegant constructions? Convoluted or unnecessarily inefficient ways to achieve some result?

Problem 2 (10 pts.)

Consider the following binary tree with 16 nodes:



Describe the order of nodes visited in each of the following traversals of the tree:

- Pre-Order
- In-Order
- Reverse In-Order
- Post-Order
- Level Order

Problem 3 (15 pts.)

Outline a method for implementing a queue where all of the elements are stored internally in two stacks.

Your answer should be in the form of a step-by-step verbal explanation or loose pseudocode; statements like "sort the list" or "find the nearest element after x that is greater than n" are acceptable, something like "move the element x to some position behind y" or "do the same thing to the opposite half" are too vague.

Problem 4 (15 pts.)

Outline a method for implementing a stack where all the elements are stored internally in a queue.

Your answer should be in the form of a step-by-step verbal explanation or loose pseudocode; statements like ”sort the list” or ”find the element after x that is greater than n are acceptable, something like ”move the element x to some position behind y” or ”do the same thing to the opposite half” are too vague.