ECS 36C: Homework #2 - Written part

Joël Porquet, UC Davis

Due before 7:00pm, Thursday, November 8th, 2018

Name: _____ **Student ID:** _____

Preamble

Goal The goal of this homework is to further your understanding of lists, stacks and queues, and trees.

Submission This document is to be filled out and uploaded to Gradescope by the due date and time.

You can either print this document, fill it out manually, and scan it back into a digital document; or you can, with a PDF editor, work directly on the digital version.

In any case, you **cannot alter** the document's formatting, as originally published. If your submission doesn't respect the expected formatting, you will be penalized.

Code of Conduct This homework is to be worked **alone**.

Each question will be entirely graded by a single TA so it is easy to notice identical or highly similar answers. Any suspicion of cheating beyond reasonable doubt, will be reported to SJA and might incur academic and disciplinary sanctions.

Lists

Question 1 ***(14 pts)*** Describe a non-recursive function for finding, by link hopping, the middle node of a doubly linked list with header and trailer pointers. You can write pseudo-code in order to further detail your answer. What is the running time of this function? *Note: This function must only use link hopping; it cannot use a counter.*

Question 2 ***(14 pts)*** Describe a fast recursive algorithm for reversing a singly linked list L, so that the ordering of the nodes becomes opposite of what it was before. You can write pseudo-code in order to further detail your answer.

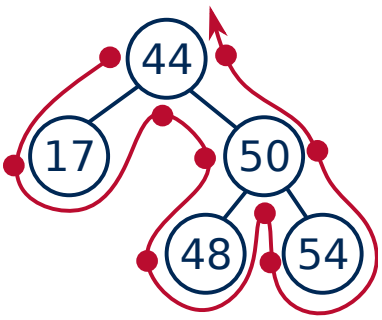
Stacks and queues

Question 3 ***(14 pts)*** Describe how you would implement a queue with two stacks so that each queue operations takes a constant amortized number of stack operations. You can write pseudo-code in order to further detail your answer. *Hint: If you push elements onto a stack and then pop them all, they appear in reverse order. If you repeat this process, they're now back in order.*

Question 4 ***(14 pts)*** Suppose you have a stack S containing n elements and a queue Q that is initially empty. Describe how you can use Q to scan S to see if it contains a certain element x , with the additional constraint that your algorithm must return the elements back to S in their original order. You may not use an array or linked list - only S and Q and a constant number of reference variables.

Trees

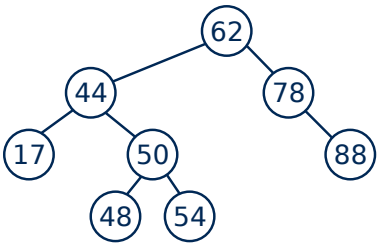
Question 5 ***(14 pts)*** A Euler tour of a binary tree is a traversal method that goes around all the nodes, just like with the pre-, in-, or post-order traversals, but processes nodes each time the path gets to them, which can happen up to three times.



Given the following tree, the Euler tour would be: `44 17 44 50 48 50 54 50 44`.

Write recursive method `void EulerRecur(Node *n)` that performs an Euler tour when called on a tree's root node. You can assume that a node has members `item` (to be printed in the method), `left` and `right`.

Question 6 ***(14 pts)*** Consider the following binary tree:



- Draw the AVL tree resulting from the insertion of an entry with key 52. Justify the rotations that needed to be performed if any.

- Draw the AVL tree resulting from the removal of entry 62. Justify the rotations that needed to be performed if any. *This part is independent from the first part.*

Question 7 ***(16 pts)*** Consider the sequence of keys (5, 16, 22, 26, 36, 50, 67, 91), and assume they are inserted in the given order in a Red-Black tree. Draw the resulting trees, after each insertion.