Algorithms

CMPT 435

Assignment 1 - 100 points

Goals

Requirements and Notes

R

A

C

E

C

Α

R

to program a few elementary data structures so we can experiment with them later on.

Develop a singly linked list.

[20 points]

Using your linked list, develop a stack. You must implement it yourself; you may not use any built-in features of the language or its libraries.

[20 points]

Using your linked list, develop a queue. You must implement it yourself; you may not use any built-in features of the language or its libraries.

[20 points]

Download the text file magicitems.txt from our web site.

[30 points]

Read it line by line into array.

Check each element of the array to see if it's a palindrome. (Ignore spaces and capitalization.) Print it if so.

To check whether or not a given string is a palindrome, take it character by character and push each on a stack and enqueue each on a queue. When every character is on a stack and in a queue, pop the stack and dequeue the queue one character at a time. If they to check for palindromes. I don't care. Do it this way.)

always match, then the string is a palindrome. (There are other ways

Create a LaTeX document that includes code listings (with line numbers) for your stack and queue. Explain how each works, referencing line numbers in the listings to be really clear.

[10 points]

1

Stack

Your code must ...

separate structure from presentation.

be professionally formatted yet uniquely yours (show some personality) $[-\infty \text{ if not}]$

use and demonstrate best practices.

make me proud to be your teacher.

Resources

- Linked lists are described in our text in chapter 10.2, starting on page EC.
- Stacks and queues are described in our text in the beginning of chapter 10, starting on page 1110 1000.

Hints

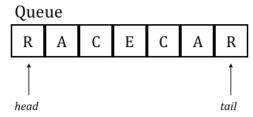
Make sure that I have approved of your programming language (the one about which you wrote a limerick in Assignment 0) before you begin.

Submitting Your Work

Make many commits to GitHub. I do not want to see one massive "everything" commit when I review your code. (It's $-\infty$ if you do that.) Commit early and often. And make sure your commit messages are descriptive, informative, and — if possible — entertaining.

Be sure that you make your final commit for this assignment on or before the due date.

(See our syllabus for those details.)



Algorithms