# DSEI1020 Fall 2019, Intro to Data Science, Michael Grossberg

## HW2:P4 Reviewing Data Structures

In this assignment you will be reviewing the kind of data Structures we did in class. You will be asked to implement something in a later problem so make sure you are comfortable. For now this assignment will require you to make 2 good multiple choice questions for all the sections I am assigning you. For this assignment you will need the student membership of the acm.org ($20/yr) and using that get to Oreilly Safari through the acm learning center and the free membership in Lynda.com via the New York Public Library

### Why learn this?

**Reason 1:** Choosing the right data structures are important both for data engineering and data science.

In CS/Engineering/Life we always have constraints. Time is limited, space is limited, computational cost can be an issue (particularly when you pay for it in the cloud). Choosing the right data structure can sometimes help or even sometime move somthing from impossible to easy.

**Reason 2:** Industry contacts are telling us that they are finding that many people interviewing for data scientists positions don't know data structures. Data structures questions are easy to ask and so they are assuming as technical people you can answer them:

* <https://www.geeksforgeeks.org/commonly-asked-data-structure-interview-questions-set-1>
* <https://medium.freecodecamp.org/the-top-data-structures-you-should-know-for-your-next-coding-interview-36af0831f5e3>

### Assignment Part A

Watch all of ["Programming Foundations: Data Structures" by Kathryn Hodge](https://www.lynda.com/Python-tutorials/Programming-Foundations-Data-Structures/751323-2.html) in lynda.com/linkedin Learning. If you have a CS background you can skip through but make sure you watch carefully and take notes on "Big O Notation", "Pros and Cons of Lists", "Pros and cond of stacks and queues", "Pros and cons of hash-based structures", "pros and cons of tree data structures". Submit:

1. screen shot showing you watched thought the course
2. from the material in the course describe when the pros and cons of
   * arrays
   * lists
   * stacks and queues
   * hash-based data structures (Key-Values)
   * trees
   * Write this up using markdown in a text files.

### Assignment Part B

Please go to Oreilly Safari though the learning center of the acm. This requires an acm.org student membership ($20/yr) but not the very expensive Safari membership. This is a great deal. Please watch the sections

* Introduction
* Fundamentals
* Ubiquitous Lists
* Pointer Structures
* Recursive Structures (through Traversal)

in the course ["Designing Data Structures in Python" by George T. Heineman](https://learning.oreilly.com/videos/designing-data-structures/9781491928622)

For each subsection (all 17) up to and including Traversal create a multiple choice question about that subsection.

#### Guide on the questions

Each of the 17 questions should cover an important idea or topic in that section. Each of the multiple choice questions should be a plausable answer. One of the 4 should be a closely realated but not-quite-right answer. Indicate the correct answer. It should go without saying that your questions should be orignal and **not** something I can find by searching web (eg. google) or from looking at a classmate.

Example:

Section: Detect Cycles in a Link List

What is the problem with a cycle in a link list?

a: it is difficult to insert a new node

b: it is difficult to delete a new node

C: iterating through the list goes on forever <- correct

d: data is over-written <- sort of correct

### Assignment Part C

After watching write a python library based on the presented code or the code from lecture that has classes from

1. Linked List
2. Stack
3. Queue
4. Trees

Submit a link to a github.com repo with this library in it. Make the library privte but share it with me (github username mdogy). I would need to see the library being built up through multiple commits.<

Keep in mind [How to Write Beautiful Python Code With PEP 8](https://realpython.com/python-pep8/), which tells how python code should be written. Don't forget to use docstrings. The way to write good docstrings (best practices) is that [used by numpy](https://docs.scipy.org/doc/numpy/docs/howto_document.html). Some best practices for [libraries are described here](https://platanios.org/assets/pdf/teaching/writing_python_libraries.pdf).