Algorithmic Programming

1. Introduction
    - What Is Algorithm Programming and Analysis?
    - Big-O Notation
    - Case Study: An Anagram Detection Example (http://interactivepython.org/courselib/static/pythonds/AlgorithmAnalysis/AnAnagramDetectionExample.html)
        - Solution 1: Checking Off
        - Solution 2: Sort and Compare
        - Solution 3: Brute Force
        - Solution 4: Count and Compare
    - Performance of Python Data Structures Compared to C
    - Lists
    - Dictionaries
2. Basic Data Structures
    - Linear Structures
    - Stack
        - Python Implementation
        - LAB: Balance Parentheses (https://www.reddit.com/r/dailyprogrammer/comments/5llkbj/2017012_challenge_298_easy_too_many_parentheses/)
    - Queue
        - Python Implementation
        - LAB: Printer Queue
    - Deque
        - Python Implementation
        - LAB: Palindrome Checker
    - List
        - Python Implementation
        - Unordered List

        - Ordered List
        - Linked List
    - Union Find
        - Python Implementation
        - Union Find By Rank
3. Recursion
    - Introduction
    - The Three Laws of Recursion
        - A recursive algorithm must have a base case.
        - A recursive algorithm must change its state and move toward the base case.
        - A recursive algorithm must call itself, recursively.
    - LAB:
        - Sierpinski Triangle
        - Tower of Hanoi
        - Exploring a Maze
    - Dynamic Programming
        - Longest Common Sunsequence Algorithm Python Implementation
4. Sorting and Searching
    - Introduction
    - The Sequential Search
        - Python Implementation of Sequential Search
    - The Binary Search
        - Python Implementation of Binary Search
    - Hashing
        - Hash Functions
        - Collision Resolution
        - Python Implementation of Hashing
    - Sorting
        - The Bubble Sort

        - The Selection Sort
        - The Insertion Sort
        - The Shell Sort
        - The Merge Sort
        - The Quick Sort
5. Trees and Tree Algorithms
    - Introduction
    - Terms and Definitions
    - Tree Traversals
    - Binary Heap Implementation
        - Binary Heap Operations
        - The Structure Property
        - The Heap Order Property
        - Heap Operations
    - Binary Search Trees
        - LAB: https://medium.com/@jamis/weekly-programming-challenge-1-55b63b9d2a1#.nbxbfoxb6
        - Balanced Binary Search Trees
    - Search Trees
        - Search Tree Implementation
    - AVL Tree Performance
        - Python Implementation
6. Graphs and Graph Algorithms
    - Introduction
    - Terms and Definitions
    - An Adjacency Matrix
        - Python Implementation
    - Breadth First Search
        - Python Implementation
    - LAB: The Knight's Tour Problem
    - Depth First Search

        - Python Implementation
    - Topological Sorting
    - Shortest Path Problems
    - Dijkstra’s Algorithm
        - Python Implementation
        - Speeding up Dijkstra