Homework 1: Big O and Unit Testing

Due: Friday, January 31 at 5pm on Canvas

Concepts: big O analysis, pseudocode, lists, unit testing

Theory

1. (5 points) Explain how you would implement list concatenation in python and analyze the runtime using big O analysis. Be sure to define any variable(s) you use in your analysis. For example, you may let a be the resize constant for a list.

2. (10 points) Subsequences of lists.

   (a) (5 points) Write pseudocode for a function that takes in a list of integers and returns the longest *strictly* increasing subsequence (as a list). For example, the longest increasing subsequence of $[1, 2, 0, 4, 8, 9, 3]$ is $[0, 4, 8, 9]$.

   (b) (5 points) Analyze the runtime of your function. Be sure to define any variable(s) you use in your analysis.

Practice

Unit testing is a part of good programming practice. As part of this course, you will use the `pytest` package, which can be installed using pip or Anaconda. Review the `pytest` documentation at `http://doc.pytest.org/en/latest/assert.html` up to the section titled "Defining your own explanation for failed assertions".

3. (10 points) Write a function that implements your algorithm from Q2 and a corresponding test function. For this assignment, you can put all your test cases into a single function. Also, be sure to include appropriate documentation in your code (e.g. docstrings and any necessary comments).

   Your tests should cover the following input scenarios: an empty list, a decreasing list (i.e. the longest increasing subsequence has length 1), an increasing list (i.e. the longest increasing subsequence is the whole list), and at least one additional test.