CPE 202 Winter 2020

Lab 1

A Note about Writing Programs: Experienced programmers know how important it is to understand the problem and develop a solution before writing any code. Why? Because this saves time and effort in the long run. Drawing pictures and writing pseudo code clarify both the problem and how to solve it. When writing pseudo-code, at first do **not** give a complete program. Give just enough detail that your intentions are clear and unambiguous, but do not provide extraneous details (complex syntax and/or variable declarations) which are **otherwise obvious**. Thinking through examples is especially important. This saves much time later in reducing rework, finding problems, etc.

Lab 1: Recursion and Python practice

1. Write an **iterative** function to find the maximum in a list of integers. (do not use library function like max or sort.) If the list is empty, it returns **None**.
2. Write a **recursive** function to reverse a string (do not use reverse library function).
3. Write a **recursive** function to search a list of integers using **binary search** along with test cases. The function returns its index if the **target** of the search is in the list else return **None**. If the list is empty, it returns **None**. Write up a short paper describing how the algorithm works step by step with a diagram in the case of "search([2,3,5,6,9], 9)". Please include the time complexity of the algorithm in Big-O notation.
4. Write a recursive function to compute the n th Fibonacci number of Fibonacci Numbers. Write up a short paper describing how the algorithm works step by step with a diagram in the case of "fib(3)". Please include the time complexity of the algorithm in Big-O notation.

Fibonacci Numbers are a sequence of numbers each of which is the sum of the two preceding ones, starting from 0 and 1. When F(n) denotes n th number of Fibonacci Numbers,

$F(0) = 0$, $F(1) = 1$ for $n=0$ and $n=1$ respectively and $F(n) = F(n-1) + F(n-2)$ for $n > 1$.

For example, the numbers from n=0 to n=10 are 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55

5. Write two functions to compute a factorial number n!, where n is an argument passed to the functions. One function should compute the factorial iteratively (use loop). Another function should compute the factorial recursively. Write a paragraph or two describing what happens when you try to compute 1000! with the two functions and why that happens. Hint: Google "recursion limit in python" on the Internet. Also, look up an entry on "stack overflow" on Wikipedia.

Details: Download the files: (Read the section below if you are new to Python/CalPoly)

- **lab1.py**
- **lab1_test_cases.py**

Complete the file "lab1.py" by implementing the five functions including docstrings for each of them, and add more tests in lab1_test_cases.py. Make sure your functions pass the tests you write.

When you are ready to submit, run the lab1_test_cases.py and make sure your implementation passes the tests before submitting your work. You are required to demo your work.

In addition to the python file, you are required to submit write-ups for #3, #4 and #5.

Test Cases

Many people tend to focus on writing code as the singular activity of a programmer, but testing is one of the most important tasks that one can perform while programming. Proper testing provides a degree of confidence in your solution. Systematic testing helps you to discover and then fix bugs (i.e., debug). Writing high quality test cases can greatly simplify the tasks of both finding and fixing bugs and, as such, **will save you time during development**. However, testing does not guarantee that your program is correct. For this part of the lab you will practice writing some simple test cases to gain experience with the unittest framework. I recommend watching the first 20 minutes or so of the following video if you are unfamiliar with testing in Python. https://www.youtube.com/watch?v=6tNS--WetLI

Submission: Zip your lab1.py, lab1_test_cases.py and the three write-ups into a zip file naming it your_calpoly_user.zip (replace the "your_calpoly_user" with your calpoly user name. Ex. tkuboi.zip) and submit the zip file to the grader: https://cpe202.toshikuboi.net/ , and also to PolyLearn. Please submit your write-ups to polylearn or hand it directly to the instructor in the class.