ECE406 **Algorithm Design & Analysis** **Winter 2025**

Assignment #1
Due Fri Jan 17 by 11:59pm
See Appendix for submission instructions

Question 1 (Big-O Notation) In each of the following, determine if $f = O(g)$, $f = \Omega(g)$, or $f = \Theta(g)$. Provide an explanation in each case.

(i) $f(n) = n^2 + 200$ and $g(n) = n^2 - 150$

(ii) $f(n) = \log_{10} n$ and $g(n) = \log_2(n^3)$

(iii) $f(n) = 10^n$ and $g(n) = n^{1/3}$

(iv) $f(n) = 7n \log(n)$ and $g(n) = n^{1.25}$

(v) $f(n) = n2^n$ and $g(n) = 3^n$

(vi) $f(n) = n$ and $g(n) = n^{1+\cos n}$

(vii) $f(n) = \log_2(n)$ and $g(n) = \log_{16}(n)$

(viii) $f(n) = 2^n$ and $g(n) = 2^{n+1}$

Question 2 (Fibonacci Sequence, exercise 0.3 from text) The Fibonacci numbers $F_0, F_1, F_2, \ldots$ are defined by the rule

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \quad \text{for all } n \geq 2. \end{aligned}$$

In this problem we confirm that the sequence grows exponentially and obtain bounds on its growth rate.

(i) Use induction to prove that $F_n \geq 2^{0.5n}$, for all $n \geq 6$.

(ii) Find a constant $0 < c < 1$ such that $F_n \leq 2^{cn}$ for all $n \geq 0$. Show that your answer is correct.

(iii) What is the largest c you can find for which $F_n = \Omega(2^{cn})$.

Question 3 (Modular Arithmetic) The following questions will help in understanding modular arithmetic:

(i) If $x \equiv x' \bmod N$ and $y \equiv y' \bmod N$, show that

$$xy \equiv x'y' \bmod N.$$

(*Hint:* Use the fact that x can be written in terms of its quotient and remainder as $qN + r$ and $x \bmod N = r$. Same for y.)

(ii) Using the first part, argue that given an integer k

$$3^k \bmod 2 = (3 \bmod 2)^k = 1.$$

(iii) What is $4^{500} \pmod{17}$? (Simplify using the properties of modulo arithmetic).

Note: Submit code for the next two questions on the LEARN Dropbox for A1 *with* your signed Academic Integrity Acknowledgement (see Appendix for details)

In this course we will be using **Python 3**. The following questions will help you get comfortable.

Question 4 (Fibonacci Warm-up) Using `A1Q4.py` as a template, complete the two functions.

(i) The inefficient Fibonacci Sequence algorithm we discussed in class.

```
def fib1(n):
    """
    Input: An integer n
    Output: The n-th number in the Fibonacci Sequence
    """
```

(ii) The efficient Fibonacci Sequence algorithm we discussed in class.

```
def fib2(n):
    """
    Input: An integer n
    Output: The n-th number in the Fibonacci Sequence
    """
```

Question 5 (Algorithms with Numbers) Using `A1Q5.py` as a template, complete the two functions.

(i) The modular exponentiation algorithm from class and the text.

```
def modexp(x, y, N):
    """
    Input: Three positive integers x and y, and N.
    Output: The number x^y mod N
    """
```

This function should return a single integer and pass the test cases that are provided in the Python script along with some others that you make up.

(ii) The extended Euclid algorithm from class and the text.

```
def extended_euclid(a, b):
    """
    Input: Two positive integers a >= b >= 0
    Output: Three integers x, y, and d returned as a tuple (x, y, d)
            such that d = gcd(a, b) and ax + by = d
    """
```

This function should return a tuple containing three integers and should pass the test cases that are provided in the Python script.

(*Hint:* In python % is the modulo operator. So $12 \,\%\, 7 = 5$. In Python 3, once you import math, math.floor returns an integer that is the floor of a float. So `math.floor`$(10.2) = 10$. You should should not require any other Python modules or special functions.)

Appendix

How do I submit this assignment? To submit the assignment, please complete the following steps.

(i) Upload your typed or handwritten work for *each* question using the ECE406 **Crowdmark** site. This includes your answers to the programming questions, where you should submit screenshots of the Python functions you have implemented.

(ii) Upload your source code for each programming question to the **A1 Dropbox submission folder** in LEARN. The starter code for each question specifies the filename convention you should follow.

(iii) Sign and upload Academic Integrity form to the **A1 Dropbox submission folder**. Use the filename `ece406w25-<WatIAMusername>-Academiic-Integrity.pdf`, where `<WatIAMusername>` is replaced by your WatIAM username (be sure to use pdf format). **Note:** you only need to submit this form once per term, typically with the first assignment. This signed form confirms that *all* your submitted assignments adhere to UW academic Integrity policies and the ECE406 course outline. If you are not sure what are signing, take a moment to read those documents.