
Project 1 - Finding the Greatest Common Divisor (GCD)

This project is due **Wednesday, November 8 at 11:59PM**. Upload a zipped file named `yourlastnameFirstinitial_proj1` to Canvas. The file must include `a(n)`:

1. Executive Summary Report in PDF format
2. Python (`.py`) file
3. `readme.txt` file with instructions on how to run your code

The main goals of this project are to:

- Come up to speed quickly on Python coding
- Gain practical experience analyzing algorithms
- Use code as a tool to conduct experiments
- Write a professional report with your findings

For this project you will implement and compare 3 algorithms for determining the greatest common divisor of two nonnegative, not both zero, integers m and n . To determine the most efficient algorithm, you will run timing experiments. Here are various levels of pseudo code for the three algorithms you will code that solve $\text{GCD}(m,n)$:

- Euclid's Algorithm

```
ALGORITHM Euclid(m, n)
//Input: Two nonnegative, not-both-zero integers m and n
//Output: Greatest common divisor of m and n
while n not equal to 0 do
    r = m mod n
    m = n
    n = r
return m
```

- Consecutive Integer Checking Algorithm

Step 1 Assign the value of $\min\{m, n\}$ to t .

Step 2 Divide m by t . If the remainder of this division is 0, go to Step 3; otherwise, go to Step 4.

Step 3 Divide n by t . If the remainder of this division is 0, return the value of t as the answer and stop; otherwise, proceed to Step 4.

Step 4 Decrease the value of t by 1. Go to Step 2.

- Middle School Procedure

Step 1 Find the prime factors of m .

Step 2 Find the prime factors of n .

Step 3 Identify all the common factors in the two prime expansions found in Step 1 and Step 2. (If p is a common factor occurring p_m and p_n times in m and n , respectively, it should be repeated $\min\{p_m, p_n\}$ times.)

Step 4 Compute the product of all the common factors and return it as the greatest common divisor of the numbers given.

Thus, for the numbers 60 and 24, we get

$$60 = 2 \cdot 2 \cdot 3 \cdot 5$$

$$24 = 2 \cdot 2 \cdot 2 \cdot 3$$

$$\gcd(60, 24) = 2 \cdot 2 \cdot 3 = 12.$$

This is a popular coding interview question. You are using code as a tool to analyze algorithms. You are graded on your code and the thoroughness of your experimentation and professionalism in your report. Here are the recommended steps to complete the project:

- Step 1. Write your algorithms in Python 3.x and test with several m, n .
- Step 2. Write a function to time the execution of each of the algorithms:

```
effGCD(s1,s2)
    Start the clock
    Call one of your GCD functions
    Stop the clock
    Print the answer and time it took
```

- Step 3. Write a user interface that inputs m, n , runs all three algorithms and times the execution, then prints out the GCD and the timing results for each. Be sure that the algorithm checks for illegal input (such as zero). If the user enters illegal input, they are asked to try again 2 more times. The program quits and says goodbye.

- Step 4. Conduct a thorough experiment to compare the time and space efficiencies of your algorithms. Choose several examples of m and n to conduct a thorough analysis.
 - Step 5. Write your professional report. In your report, include the following:
 - A one paragraph executive summary (how you conducted your experiment, lessons learned, what are the most and least efficient (time and space) algorithms and why).
 - A table that shows the results of 10 test runs of different m , n input. The columns of the table are: m,n , Euclid Runtime, Integer Checking Runtime, Middle School Procedure Runtime, GCD, time efficiency, space efficiency, your comments. Please include $m=31415$ and $n=14142$ as your first table entry.
-