Lab Notes:

Euclid's algorithm is a method to compute the greatest common divisor (GCD) of two integers. The GCD of two numbers is the largest number that divides both without leaving a remainder.

**Algorithm**:

The algorithm is based on the observation that the GCD of two numbers doesn't change if the larger number is replaced by its remainder when divided by the smaller number. This process is repeated until one of the numbers becomes zero. At that point, the GCD is the other number.

**Steps**:

1. Given two numbers **`a`** and **`b`,** where (a > b), we want to find `GCD (a, b) `.

2. Compute the remainder of **`a`** divided by **`b`**, i.e., `r = a % b`.

3. Replace **`a`** with **`b`** and **`b`** with **`r`.**

4. Repeat the process until **`b`** becomes zero. The GCD will be the value of **`a`** at that point.

**Example**:

Let’s find the GCD of 56 and 98:

1. (a = 98), (b = 56)

2. (98 mod 56 = 42) (remainder)

3. Now, replace (a = 56) and (b = 42)

4. (56 mod 42 = 14)

5. Now, replace (a = 42) and (b = 14)

6. (42 mod 14 = 0)

Since (b = 0), the GCD is (a = 14).

A computer screen shot of a number

Description automatically generated

result = gcd (98, 56)  
print (f"The GCD of 98 and 56 is: {result}")

This algorithm is very efficient and works even for large numbers!

**Lecture Notes:**

Largest natural number that divides both m, n 🡪 GCD (m, n)

If one of the numbers for example from m or n are negative, we can ignore as it’s considered the same.

In this lecture we are assuming that m and n are not 0

GCD (10,15) is equal to GCD (15,10)

**Euclid’s Algorithm**

The idea is that the GCD of two number m and n have the following cases

m if n = 0  
GCD (n, m mod n), the term mod represents, the remainder when m is divided by n  
GCD (87,15) = GCD (15, 87 mod 15) 87 = 15\*5+12’  
GCD (87,15) = GCD (15, 12) = GCD (12, 15 mod 12) = GCD (12, 3)  
GCD (12, 3) = GCD (3,0) = 3 ----- (Base Case)

Algorithm of Euclid's algorithm

A person writing on a blackboard

Description automatically generated

The complexity of their algorithm:  
  
This is a lucky case, where we were able to find the GCD in one step  
A hand writing on a blackboard

Description automatically generated  
  
Worse Case:  
The way to measures the complexity is not on the magnitude of the numbers (m,n) but the number of bits (size of the input) that are needed to represent m and n

Fibonacci numbers is where each number of bits is the sum of previous numbers

Assignment:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated