INDIAN INSTITUTE OF TECHNOLOGY GOA

Name: Aditya Rajesh Bawangade

Roll Number: 2103111

The output of the program when ran for 6/10:

```
[Running] cd "d:\LECS AND MATERIAL\SEMESTER 4\Algorithm Design\Assignment6\" && g++
noModNDiv_2103111.cpp -o noModNDiv_2103111 && "d:\LECS AND MATERIAL\SEMESTER 4\Algorithm
Design\Assignment6\"noModNDiv_2103111
terminate called after throwing an instance of 'std::runtime_error'
  what():  This is a runtime error as you tried dividing by zero 🙄🙄🙄

[Done] exited with code=3221226505 in 1.154 seconds
```

The output of the program when ran for 6/31:

```
[Running] cd "d:\LECS AND MATERIAL\SEMESTER 4\Algorithm Design\Assignment6\" && g++
noModNDiv_2103111.cpp -o noModNDiv_2103111 && "d:\LECS AND MATERIAL\SEMESTER 4\Algorithm
Design\Assignment6\"noModNDiv_2103111
6

[Done] exited with code=0 in 0.724 seconds
```

The analysis of the Extended Euclid Algorithm implemented for this assignment is as follows:

The provided code carries out the **Extended Euclidean method**, which determines the greatest common factor of two numbers as well as a pair of coefficients **(x, y)** that make **ax + by = gcd (a, b)**. The code's bit-level time analysis is shown here:

The function accepts as input two numbers with bit lengths of **n**: **a** and **b**.

When b is zero, the function initialises and returns a pair of integers and a pair of pairs in **O(1)** time using constant time operations.

In the recursive instance, the function divides first using the divide function, which requires an **O(n^2)** amount of time.

The function then repeatedly calls itself with the inputs **b** and **a mod b**.

The function initialises two pairs of integers and two pairs of pairs on the next line using constant time operations. The function then computes and returns the outcome in O(1) time using constant time operations. Due to the division operation in step 3, the function's overall bit-level time complexity is **O(n^2)** or for better generalisation, we may call this as **<u>O(log(a)*log(b))</u>**.