Euclid's Algorithm Explained

■ Step 1: The setup

We want to compute gcd(m, n) (the greatest common divisor of two numbers m and n). If n divides m (i.e., remainder is 0), then gcd(m, n) = n. If not, we break down m in terms of n.

■ Step 2: Division form

If n does not divide m, then by the division rule: $m = qn + r$ where q = quotient and r = remainder ($0 \le r < n$).

■ Step 3: Factorization

Suppose d is a common divisor of m and n. So we can write: $m = ad$, $n = bd$. Now substitute into the division equation: $m = qn + r \rightarrow ad = q(bd) + r \rightarrow r = (a - qb)d$. ■ This shows that r is also a multiple of d. That means any common divisor of m and n also divides r. So: gcd(m, n) = gcd(n, r).

■ Step 4: The recursive rule

Since r = m mod n, the algorithm becomes: gcd(m, n) = gcd(n, m mod n). And if remainder = 0, gcd is n.

■ Step 5: Python code in notes

```
def gcd(m, n):
    (a, b) = (max(m, n), min(m, n))  # ensure a >= b
    if a % b == 0:   # if b divides a exactly
        return b
    else:
        return gcd(b, a % b)  # recursion with remainder
```

■ Example

Find gcd(48, 18):

```
gcd(48, 18)
48 % 18 = 12 → gcd(18, 12)

gcd(18, 12)
18 % 12 = 6 → gcd(12, 6)

gcd(12, 6)
12 % 6 = 0 → return 6 ■

So gcd(48, 18) = 6.
```

■ In short:

• Write $m = qn + r$ • Replace (m, n) with (n, r) • Keep going until $r = 0$ • The answer is the last non-zero divisor.