Advanced Number Theory

- ✔ Euclidean Algorithm
- ✔ GCD Properties
- ✔ Euler’s Totient Function
- ✔ Euler’s Theorem
- ✔ Fermat’s Theorem
- ✔ Mod Inverse under Euler’s & Fermat’s Theorem

Euclidean Algorithm: Link

Theorem:

$$\gcd(a,b) = \begin{cases} a, & \text{if } b = 0 \\ \gcd(b, a \bmod b), & \text{otherwise.} \end{cases}$$

Implementation:

Recursive

```
int gcd (int a, int b) {
    if (b == 0)
        return a;
    else
        return gcd (b, a % b);
}
```

Iterative

```
int gcd (int a, int b) {
    while (b) {
        a %= b;
        swap(a, b);
    }
    return a;
}
```

Time Complexity: O(log(min(a, b))) Proof

Important GCD results

- GCD(a, b) = GCD(b, a)
- GCD(a, 0) = a
- GCD(a, b, c) = GCD(GCD(a, b), c) = GCD(a, GCD(b, c)) = GCD(b, GCD(a, c))
- GCD(a, b) >= GCD(a, b, c) >= GCD(a, b, c, d)
- GCD contains the minimum powers of primes
- LCM contains the maximum powers of primes
- GCD(a, b) * LCM(a, b) = a * b

Euler's Totient Function: Link to study further

phi(N) = number of values X such that X <= N and gcd(X, N) = 1

phi(N) is a multiplicative function.

Properties

The following properties of Euler totient function are sufficient to calculate it for any number:

- If p is a prime number, then $\gcd(p, q) = 1$ for all $1 \leq q < p$. Therefore we have:

$$\phi(p) = p - 1.$$

- If p is a prime number and $k \geq 1$, then there are exactly p^k / p numbers between 1 and p^k that are divisible by p. Which gives us:

$$\phi(p^k) = p^k - p^{k-1}.$$

- If a and b are relatively prime, then:

$$\phi(ab) = \phi(a) \cdot \phi(b).$$

Euler’s Totient Function:

Idea:

$$\phi(n) = \phi({p_1}^{a_1}) \cdot \phi({p_2}^{a_2}) \cdots \phi({p_k}^{a_k})$$

$$= \left({p_1}^{a_1} - {p_1}^{a_1-1}\right) \cdot \left({p_2}^{a_2} - {p_2}^{a_2-1}\right) \cdots \left({p_k}^{a_k} - {p_k}^{a_k-1}\right)$$

$$= p_1^{a_1} \cdot \left(1 - \frac{1}{p_1}\right) \cdot p_2^{a_2} \cdot \left(1 - \frac{1}{p_2}\right) \cdots p_k^{a_k} \cdot \left(1 - \frac{1}{p_k}\right)$$

$$= n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right)$$

Euler’s Totient Function:

Implementation:

```
int phi(int n) {
    int result = n;
    for (int i = 2; i * i <= n; i++) {
        if (n % i == 0) {
            while (n % i == 0)
                n /= i;
            result -= result / i;
        }
    }
    if (n > 1)
        result -= result / n;
    return result;
}
```

Euler’s Totient Function:

Bonus problem:

- Find phi(x) for all numbers from 1 to N. Link

Nice property:

- Divisor sum property. Link

Euler's Theorem and Fermat's Theorem: Link

Euler's Theorem $a^{\phi(m)} \equiv 1 \pmod{m}$ if a and m are relatively prime.

Fermat's Theorem $a^{m-1} \equiv 1 \pmod{m}$

Practice Problem: Link

Mod Inverse using Euler's Theorem and Fermat's Theorem:

$$a^{\phi(m)} \equiv 1 \mod m \longrightarrow a^{\phi(m)-1} \equiv a^{-1} \mod m$$

$$a^{m-1} \equiv 1 \mod m \longrightarrow a^{m-2} \equiv a^{-1} \mod m$$

Not so important stuff (Learn on your own)

- Extended Euclidean Algo Link
- Linear Diophantine Equations Link
- Mod Inverse in O(logM) for general M Link
- Chinese Remainder Theorem Link
- Checking for Prime in O(logP) - Miller Rabin Test Link