Euler's totient function

The Euler's Totient Function, also known as phi-function denoted as $\varphi(n)$ (that's the Greek letter "phi"), counts the number of integers from 1 to n that are coprime with n (i.e., their greatest common divisor (GCD) with n is 1).

Example:

For n = 6:

Numbers from 1 to 6: 1, 2, 3, 4, 5, 6

GCD(1,6)=1, GCD(2,6)=2, GCD(3,6)=3, GCD(4,6)=2, GCD(5,6)=1, GCD(6,6)=6

Numbers coprime with 6: 1 and 5

So $\varphi(6) = 2$

$\varphi(1) = 1$ → {1}

$\varphi(2) = 1$ → {1}

$\varphi(3) = 2$ → {1,2}

$\varphi(4) = 2$ → {1,3}

$\varphi(5) = 4$ → {1,2,3,4}

$\varphi(6) = 2$ → {1,5}

$\varphi(7) = 6$ → {1,3,5,7}

$\varphi(8) = 4$ → {1,2,4,5,7,8}

$\varphi(12) = 4$ → {1,5,7,11}

Properties of Euler's Totient Function

1. For a prime number p: $\varphi(p) = p-1$

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

$$\varphi(p) = p-1$$

Example: $\varphi(7) = 6$ (since 1,2,3,4,5,6 are all coprime with 7)

2. For a power of a prime p^k: $\varphi(p^k) = p^k - p^{k-1}$

↳ The only numbers that are not coprime with p^k are the multiples of p (since p is the only prime factor of p^k)

↳ So we can calculate $\varphi(p^k)$ by:

-> Start with all numbers from 1 to p^k (total = p^k numbers)

-> Subtract the numbers that are not coprime with p^k (i.e., multiples of p)

↳ Between 1 and p^k, the multiples of p are:
$p, 2p, 3p, ..., p^{k-1}p$ (which is p^k)

Total multiples = p^{k-1}

Therefore:

$\varphi(p^k)$ = (Total numbers) - (Non-coprime numbers)
$= p^k - p^{k-1}$

↳ Example: $\varphi(8) = \varphi(2^3)$

Total numbers: 1,2,3,4,5,6,7,8 → 8 numbers

Non-coprime numbers (multiples of 2): 2,4,6,8 → 4 numbers

$\varphi(8) = 8 - 4 = 4$

Coprime numbers: 1,3,5,7

3. Multiplicative property:

If two numbers a and b are coprime ($\gcd(a,b)=1$), then:
$\varphi(a*b) = \varphi(a) * \varphi(b)$

Example: $\varphi(6) = \varphi(2)*\varphi(3) = 1*2 = 2$ (which matches our first example)

4. Gauss's Divisor Sum Property:

For any positive integer n,
The sum of $\varphi(d)$ over all positive divisors d of n equals n itself.

Mathematically:

$$\sum_{d|n} \varphi(d) = n$$

Where:

$d|n$ means "d divides n" (all positive divisors of n)

$\varphi(d)$ is Euler's totient function for each divisor d

Example: n = 10

Divisors of 10: 1, 2, 5, 10

Calculate φ for each:

$\varphi(1) = 1$ (only number is 1, $\gcd(1,1)=1$)

$\varphi(2) = 1$ (number: 1)

$\varphi(5) = 4$ (numbers: 1,2,3,4)

$\varphi(10) = 4$ (numbers: 1,3,7,9)

Sum: $\varphi(1) + \varphi(2) + \varphi(5) + \varphi(10) = 1 + 1 + 4 + 4 = 10$ ✓

Formula for Euler's Totient Function

Thus, using the first three properties, we can compute $\varphi(n)$ through the factorization of n (decomposition of n into a product of its prime factors).

If , where $n = p_1^{a_1} \cdot p_2^{a_2} \cdots p_k^{a_k}$ p_i are prime factors of n,

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

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

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

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

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

Calculating Euler's Totient Function

1. Basic Implementation (for single value)

Here is an implementation using factorization in $O(\sqrt{n})$:

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

2. Precompute Totient Values for All Numbers (Sieve Approach)

This is efficient when you need totient values for many numbers:

```
vector<int> computeTotients(int max_n) {
    vector<int> phi(max_n + 1);
    for (int i = 0; i <= max_n; i++)
        phi[i] = i;

    for (int p = 2; p <= max_n; p++) {
        if (phi[p] == p) { // p is prime
            phi[p] = p - 1;
            for (int multiple = 2 * p; multiple <= max_n; multiple += p) {
                phi[multiple] = (phi[multiple] / p) * (p - 1);
            }
        }
    }
    return phi;
}
```

For each multiple of p (2p, 3p, 4p, ... ≤ max_n):

Updates φ(multiple) using the formula: φ(multiple) = (φ(multiple)/p) * (p-1)

This works because:

When we first encounter a multiple, φ(multiple) = multiple

For each prime factor p, we adjust by multiplying by (1 - 1/p) = (p-1)/p

φ(multiple) = multiple * Π(1 - 1/p) for all distinct prime factors p

TC: O(n log log (n))

Euler's Theorem and Its Applications:

Euler's theorem is a fundamental result in number theory that connects Euler's totient function with modular arithmetic.

Euler's Theorem Statement

For any two positive integers a and n that are coprime ($\gcd(a, n) = 1$):

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

Where:

$\varphi(n)$ is Euler's totient function (count of numbers ≤ n that are coprime with n)

≡ denotes congruence modulo n

Special Case: Fermat's Little Theorem

When n is prime (let's denote it as p):

$\varphi(p) = p - 1$

So Euler's theorem becomes:

$$a^{p-1} \equiv 1 \pmod{p}$$

This is Fermat's little theorem.

Key Applications

1. Modular Exponentiation Simplification

From Euler's theorem, we get the equivalence:

$$a^n \equiv a^{n \bmod \varphi(m)} \pmod{m} \quad \text{(when } \gcd(a, m) = 1\text{)}$$

This is incredibly useful for computing large exponents modulo m.

Example: Compute 3^{100} mod 10

$\gcd(3, 10) = 1$, so Euler's theorem applies

$\varphi(10) = \varphi(2)*\varphi(5) = 1*4 = 4$

100 mod 4 = 0

Therefore, $3^{100} \equiv 3^0 \equiv 1$ mod 10

2. Computing Modular Multiplicative Inverses

If $\gcd(a, m) = 1$, then the inverse of a modulo m exists and can be found as:

$$a^{-1} \equiv a^{\varphi(m)-1} \pmod{m}$$

Example: Find the inverse of 3 modulo 7

$\varphi(7) = 6$

$3^{(-1)} = 3^{(6-1)} = 3^5$ mod 7

$3^5 = 243 = 243 - 3*49 = 243 - 238 = 5$ mod 7

Verification: 3*5 = 15 ≡ 1 mod 7 ✓