GUÍA DE APRENDIZAJE PARA CONCURSANTES ICPC Y IOI: LA FUNCIÓN TOTIENTE DE EULER ($\phi(n)$)

1. Introducción

La función totient de Euler, también conocida como función phi $\phi(n)$, cuenta el número de números enteros entre 1 y n inclusive, que son coprimos de n.

2. Conocimientos previos

2.1. Números coprimos

Dos números son coprimos si su máximo común divisor es 1 (1 se considera coprimo para cualquier número).

2.2. Leonhard Euler

Leonhard Euler fue un matemático y físico suizo. Se trata del principal matemático del siglo XVIII y uno de los más grandes y prolíficos de todos los tiempos, muy conocido por el número de Euler (e), número que aparece en muchas fórmulas de cálculo y física.

2.3. Criba de Eratosthenes

La Criba de Eratóstenes es un algoritmo utilizado para encontrar todos los números primos hasta un cierto límite dado. Fue desarrollado por el matemático griego Eratóstenes en el siglo III a.C. y es uno de los métodos más antiguos y eficientes para encontrar números primos.

3. Desarrollo

Aquí están los valores de $\phi(n)$ para los primeros números enteros positivos:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
$\phi(n)$	1	1	2	2	4	2	6	4	6	4	10	4	12	6	8	8	16	6	18	8	12

Propiedades:

Las siguientes propiedades de la función totiente de Euler son suficientes para calcularla para cualquier número:

1. Si p es un número primo, entonces $\gcd(p, q) = 1$ para todo $1 \leq q < p$. Por lo tanto tenemos:

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

2. Si p es un número primo y $k \geq 1$, entonces hay exactamente p^k/p números entre 1 y p^k que son divisibles por p . Lo que nos da:

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

3. Si a y b son primos relativos, entonces:

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

   Esta relación no es trivial de ver. Se deduce del teorema del resto chino . El teorema del resto chino garantiza que para cada $0 \leq x < a$ y cada $0 \leq y < b$, existe un $0 \leq z < ab$ único con $z \equiv x \pmod{a}$ y $z \equiv y \pmod{b}$. No es difícil demostrar que z es coprimo con ab si y solo si x es coprimo con a y y es coprimo con b . Por lo tanto, la cantidad de números enteros coprimos de ab es igual al producto de las cantidades de a y b .

4. En general, para a y b no coprimos , la ecuación:

$$\phi(ab) = \phi(a) \cdot \phi(b) \cdot \frac{d}{\phi(d)}$$

   con $d = \gcd(a, b)$ se mantiene.

Por lo tanto, usando las primeras tres propiedades, podemos calcular $\phi(n)$ mediante la factorización de n (descomposición de n en un producto de sus factores primos). Si $n = {p_1}^{a_1} \cdot {p_2}^{a_2} \cdots {p_k}^{a_k}$, donde p_i son factores primos de n.

$$\begin{aligned}
\phi(n) &= \phi({p_1}^{a_1}) \cdot \phi({p_2}^{a_2}) \cdots \phi({p_k}^{a_k}) && (1)\\
&&& (2)\\
&= \left({p_1}^{a_1} - {p_1}^{a_1-1}\right) \cdot \left({p_2}^{a_2} - {p_2}^{a_2-1}\right) \cdots \left({p_k}^{a_k} - {p_k}^{a_k-1}\right) && (3)\\
&&& (4)\\
&= p_1^{a_1} \cdot \left(1 - \frac{1}{p_1}\right) \cdot p_2^{a_2} \cdot \left(1 - \frac{1}{p_2}\right) \cdots p_k^{a_k} \cdot \left(1 - \frac{1}{p_k}\right) && (5)\\
&&& (6)\\
&= n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right) && (7)
\end{aligned}$$

3.1. Función del totiente de Euler de 1 a n

Si necesitamos todos los tocientes de todos los números entre 1 y n , entonces factorizar todos los n números no es eficiente. Podemos utilizar la misma idea que la Criba de Eratóstenes . Todavía se basa en la propiedad que se muestra arriba, pero en lugar de actualizar el resultado temporal de cada factor primo para cada número, encontramos todos los números primos y para cada uno actualizamos los resultados temporales de todos los números que son divisibles por ese número primo.

3.2. Propiedad de la suma divisoria

Esta interesante propiedad fue establecida por Gauss:

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

Aquí la suma es sobre todos los divisores positivos d de n .

Por ejemplo, los divisores de 10 son 1, 2, 5 y 10. Por lo tanto, $\phi(1) + \phi(2) + \phi(5) + \phi(10) = 1. + 1 + 4 + 4 = 10$.

3.2.1. Encontrar el totiente del 1 al n usando la propiedad de la suma del divisor

La propiedad de la suma del divisor también nos permite calcular el tociente de todos los números entre 1 y n . Esta implementación es un poco más simple que la implementación anterior basada en la criba de Eratóstenes, sin embargo también tiene una complejidad ligeramente peor: O$(n \log n)$

3.3. Aplicación en el teorema de Euler

La propiedad más famosa e importante de la función totiente de Euler se expresa en el teorema de Euler :

$$a^{\phi(m)} \equiv 1 \pmod{m} \quad \text{si } a \text{ y } m \text{ son primos relativos.}$$

En el caso particular en el que m es primo, el teorema de Euler se convierte en el pequeño teorema de Fermat :

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

El teorema de Euler y la función totiente de Euler ocurren con bastante frecuencia en aplicaciones prácticas; por ejemplo, ambos se utilizan para calcular el inverso multiplicativo modular .

Como consecuencia inmediata también obtenemos la equivalencia:

$$a^n \equiv a^{n \bmod \phi(m)} \pmod{m}$$

Esto permite calcular $x^n \bmod m$ para n muy grandes , especialmente si n es el resultado de otro cálculo, ya que permite calcular n bajo un módulo.

3.3.1. Teoría de grupos

$\phi(n)$ es el orden del grupo multiplicativo mod n $(\mathbf{Z}/n\mathbf{Z})^\times$, es decir el grupo de unidades (elementos con inversos multiplicativos). Los elementos con inversos multiplicativos son precisamente aquellos coprimos de n .

El orden multiplicativo de un elemento a mod n , denotado $\mathrm{ord}_n(a)$, es el $k > 0$ más pequeño tal que $a^k \equiv 1 \pmod{m}$. $\mathrm{ord}_n(a)$ es el tamaño del subgrupo generado por a , por lo que según el teorema de Lagrange, el orden multiplicativo de cualquier a debe dividir $\phi(n)$. Si el orden multiplicativo de a es $\phi(n)$, el mayor posible, entonces a es una raíz primitiva y el grupo es cíclico por definición.

3.4. Generalización

Existe una versión menos conocida de la última equivalencia, que permite calcular $x^n \bmod m$ de manera eficiente para x y m no coprimos . Para x, m y $n \geq \log_2 m$ arbitrarios :

$$x^n \equiv x^{\phi(m)+[n \bmod \phi(m)]} \mod m$$

Prueba:

Sean $p_1, \ldots, p_t$ divisores primos comunes de x y m , y k_i sus exponentes en m . Con ellos definimos $a = p_1^{k_1} \ldots p_t^{k_t}$, lo que hace que $\frac{m}{a}$ sea coprimo con x . Y sea k el número más pequeño tal que a divida a x^k . Suponiendo $n \geq k$, podemos escribir:

$$\begin{aligned}
x^n \bmod m &= \frac{x^k}{a} a x^{n-k} \bmod m && (8)\\
&= \frac{x^k}{a} \left(a x^{n-k} \bmod m \right) \bmod m && (9)\\
&= \frac{x^k}{a} \left(a x^{n-k} \bmod a\frac{m}{a} \right) \bmod m && (10)\\
&= \frac{x^k}{a} a \left(x^{n-k} \bmod \frac{m}{a} \right) \bmod m && (11)\\
&= x^k \left(x^{n-k} \bmod \frac{m}{a} \right) \bmod m && (12)
\end{aligned}$$

La equivalencia entre la tercera y cuarta línea se deriva del hecho de que $ab \bmod ac = a(b \bmod c)$. De hecho, si $b = cd + r$ con $r < c$, entonces $ab = acd + ar$ con $ar < ac$.

Dado que x y $\frac{m}{a}$ son coprimos, podemos aplicar el teorema de Euler y obtener la fórmula eficiente (ya que k es muy pequeña; de hecho, $k \leq \log_2 m$):

$$x^n \bmod m = x^k \left(x^{nk \bmod \phi(\frac{m}{a})} \bmod \frac{m}{a} \right) \bmod m.$$

Esta fórmula es difícil de aplicar, pero podemos usarla para analizar el comportamiento de $x^n \bmod m$. Podemos ver que la secuencia de potencias $(x^1 \bmod m, x^2 \bmod m, x^3 \bmod m, \ldots)$ entra en un ciclo de longitud $\phi\left(\frac{m}{a}\right)$ después de los primeros k (o menos) elementos. $\phi\left(\frac{m}{a}\right)$ divide $\phi(m)$ (porque a y $\frac{m}{a}$ son coprimos, tenemos $\phi(a) \cdot \phi\left(\frac{m}{a}\right) = \phi(m)$), por lo tanto también podemos decir que el período tiene longitud $\phi(m)$. Y dado que $\phi(m) \geq \log_2 m \geq k$, podemos concluir la fórmula deseada, mucho más simple:

$$x^n \equiv x^{\phi(m)} x^{(n-\phi(m)) \bmod \phi(m)} \bmod m \equiv x^{\phi(m)+[n \bmod \phi(m)]} \mod m.$$

4. Implementación

4.1. C++

4.1.1. Función totiente de Euler

```
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;
}
```

4.1.2. Encontrar el totiente del 1 al n usando la propiedad de la suma del divisor

```
void phi_1_to_n(int n) {
   vector<int> phi(n + 1);
   for (int i = 0; i <= n; i++) phi[i] = i;
   for (int i = 2; i <= n; i++) {
      if(phi[i] == i) {
         for (int j = i; j <= n; j += i) phi[j] -= phi[j] / i;
      }
   }
}
```

4.2. Java

4.2.1. Función totiente de Euler

```
public static 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;
}
```

4.2.2. Encontrar el totiente del 1 al n usando la propiedad de la suma del divisor

```
public static int [] phi_1_to_n(int n) {
   int [] phi = new int [n + 1];
   for (int i = 0; i <= n; i++) phi[i] = i;
   for (int i = 2; i <= n; i++) {
      if(phi[i] == i) {
         for (int j = i; j <= n; j += i) phi[j] -= phi[j] / i;
      }
   }
   return phi;
}
```

5. Aplicaciones

La función phi de Euler, denotada como $\phi(n)$, es una función aritmética que cuenta la cantidad de números enteros positivos menores o iguales a n que son coprimos con n (es decir, que no comparten ningún factor primo con n). Algunas aplicaciones de la función phi de Euler son:

1. **Teorema de Euler:** La función phi de Euler está relacionada con el teorema de Euler, que establece que $a^{\phi(n)} \equiv 1(\mod n)$ para todo entero a coprimo con n.
2. **Criptografía:** La función phi de Euler es utilizada en criptografía para generar claves públicas y privadas en sistemas criptográficos basados en el algoritmo RSA.
3. **Teoría de números:** La función phi de Euler es fundamental en la teoría de números y se utiliza en diversos problemas relacionados con la factorización de enteros, congruencias y otros conceptos matemáticos.
4. **Generación de números primos:** La función phi de Euler se utiliza en la generación de números primos y en la implementación de algoritmos eficientes para encontrar primos relativos a un número dado.

6. Complejidad

La función del totiente de Euler en su implementacion utiliza la factorización por tanto su complejidad será de O($\sqrt{n}$). El caso de encontrar el totiente del 1 al n usando la propiedad de la suma del divisor su implementación es similar a la criba de Eratosthenes por tanto su complejidad será de O($n \log \log n$).

7. Ejercicios

A continuación un grupo de ejercicios que se pueden resolver aplicando los contenidos relacionados con la función total de Euler:

- DMOJ - Cuban Roulette
- SPOJ 4141 Euler Totient Function
- SPOJ - LCM Sum
- SPOJ - GCDEX
- SPOJ - Totient in Permutation
- SPOJ - Totient Extreme
- SPOJ - Playing with GCD
- SPOJ - G Force
- SPOJ - Smallest Inverse Euler Totient Function
- Codeforces - The Holmes Children
- LeetCode - 372. Super Pow
- Kattis - Exponial
- Codeforces - Power Tower
- LOJ - Mathematically Hard
- UVA 12995 - Farey Sequence
- UVA 13132 - Laser Mirrors
- GYM - Simple Calculations (F)
- Codechef - Golu and Sweetness
- UVA 10990 - Another New Function
- TIMUS 1673 Admission to Exam
- UVA 11327 Enumerating Rational Numbers
- UVA 10299 Relatives
- UVA 10179 Irreducible Basic Fractions