

Algoritmos en teoría de números

IIC2283

Para recordar: aritmética modular

Dados dos números $a, b \in \mathbb{Z}$, si $b > 0$ entonces existen $\alpha, \beta \in \mathbb{Z}$ tales que $0 \leq \beta < b$ y

$$a = \alpha \cdot b + \beta$$

Además, estos números α, β son únicos

β es llamado el resto de la división entera entre a y b , y es denotado como $a \bmod b$

- Por ejemplo, $8 \bmod 3 = 2$, $9 \bmod 3 = 0$ y $(-8) \bmod 3 = 1$

Para recordar: aritmética modular

Definición

$b \equiv c \pmod{n}$ si n divide a $(c - b)$

Usamos la notación $n|m$ para indicar que n divide a m

► $b \equiv c \pmod{n}$ si $n|(c - b)$

Para recordar: algunas propiedades básicas

Proposición

1. $a \equiv b \pmod n$ si y sólo si $a \pmod n = b \pmod n$
2. $a \equiv (a \pmod n) \pmod n$
3. Si $a \equiv b \pmod n$ y $c \equiv d \pmod n$, entonces:

$$\begin{array}{rcl} (a + c) & \equiv & (b + d) \pmod n \\ (a \cdot c) & \equiv & (b \cdot d) \pmod n \end{array}$$

Ejercicios

1. Demuestre la proposición
2. Demuestre que un número n es divisible por 3 si y sólo si la suma de sus dígitos es divisible por 3

Vamos a estudiar tres algoritmos fundamentales en el área:

- ▶ Exponenciación rápida
- ▶ El algoritmo de Euclides para el cálculo del máximo común divisor
- ▶ El algoritmo de Euclides extendido y el cálculo del inverso modular

Exponenciación rápida: calculando $a^b \bmod n$

Utilizamos el siguiente algoritmo para calcular $a^b \bmod n$, el cual es llamado exponenciación rápida:

```
EXP( $a, b, n$ )  
  if  $b = 1$  then return  $a \bmod n$   
  else if  $b$  es par then  
     $val := \mathbf{EXP}(a, \frac{b}{2}, n)$   
    return  $(val \cdot val) \bmod n$   
  else  
     $val := \mathbf{EXP}(a, \frac{b-1}{2}, n)$   
    return  $(val \cdot val \cdot a) \bmod n$ 
```

Ejercicio

Considerando la multiplicación de enteros y el cálculo de la función $x \bmod y$ como las operaciones básicas a contar, demuestre que **EXP**(a, b, n) en el peor caso es $O(\log_2(b))$

Máximo común divisor

Sea $\text{MCD}(a, b)$ el máximo común divisor de los números a y b

- ¿Cómo podemos calcular $\text{MCD}(a, b)$?

Proposición

Si $b > 0$, entonces $\text{MCD}(a, b) = \text{MCD}(b, a \bmod b)$

Demostración: Vamos a demostrar que un número c divide a a y b si y sólo si c divide a b y $a \bmod b$

- De esto se concluye que $\text{MCD}(a, b) = \text{MCD}(b, a \bmod b)$

Máximo común divisor

Sabemos que $a = \alpha \cdot b + (a \bmod b)$

$(\Rightarrow)$ Suponga que $c|a$ y $c|b$.

Dado que $(a \bmod b) = a - \alpha \cdot b$, concluimos que $c|(a \bmod b)$.

$(\Leftarrow)$ Suponga que $c|b$ y $c|(a \bmod b)$.

Dado que $a = \alpha \cdot b + (a \bmod b)$, tenemos que $c|a$



Cálculo de máximo común divisor

De lo anterior, concluimos la siguiente identidad para $a > 0$:

$$\text{MCD}(a, b) = \begin{cases} a & b = 0 \\ \text{MCD}(b, a \bmod b) & b > 0 \end{cases}$$

Usamos esta identidad para generar un algoritmo para calcular el máximo común divisor, el cual es conocido como Algoritmo de Euclides:

```
MCD( $a, b$ )  
  if  $a = 0$  and  $b = 0$  then return error  
  else if  $a = 0$  then return  $b$   
  else if  $b = 0$  then return  $a$   
  else if  $a \geq b$  then return MCD( $b, a \bmod b$ )  
  else return MCD( $a, b \bmod a$ )
```

¿Cuál es la complejidad del algoritmo?

La complejidad del algoritmo

Lema

Si $a \geq b$ y $b > 0$, entonces $(a \bmod b) < \frac{a}{2}$

Demostración: Si $b > \frac{a}{2}$:

$$a \bmod b = a - b < a - \frac{a}{2} = \frac{a}{2}$$

La complejidad del algoritmo

Si $b < \frac{a}{2}$, entonces:

$$a \bmod b < b < \frac{a}{2}$$

Si $b = \frac{a}{2}$ (a debe ser par):

$$a \bmod b = 0 < b = \frac{a}{2}$$



Ejercicio

Suponga que la operación básica para el algoritmo **MCD** es el cálculo de la función $x \bmod y$. Muestre entonces que el algoritmo en el peor caso es $O(\log_2(\max\{a, b\}))$, suponiendo que la entrada es (a, b)

- Vale decir, **MCD** es de orden lineal en el tamaño de la entrada en el peor caso

Una noción importante: inverso modular

Definición

b es inverso de a en módulo n si $a \cdot b \equiv 1 \pmod{n}$

Ejemplo

37 es inverso de 13 en módulo 60

¿Todo número tiene inverso modular?

- ▶ No, 2 no tiene inverso en módulo 4

¿Bajo qué condiciones a tiene inverso en módulo n ?

Identidad de Bézout

Para cada $a, b \in \mathbb{N}$ tales que $a \neq 0$ o $b \neq 0$, existen $s, t \in \mathbb{Z}$ tales que:

$$\text{MCD}(a, b) = s \cdot a + t \cdot b$$

Demostración: Sea

$$D = \{n \in \mathbb{N} \mid n > 0 \text{ y existen } s, t \in \mathbb{Z} \text{ tal que } n = s \cdot a + t \cdot b\}$$

Dado que $a \neq 0$ o $b \neq 0$, se tiene que $D \neq \emptyset$ y, por lo tanto, D tiene un menor elemento

- Sea d este elemento, y suponga que $d = s_0 \cdot a + t_0 \cdot b$

Una identidad útil

Existen $\alpha, \beta \in \mathbb{N}$ tales que $0 \leq \beta < d$ y $a = \alpha \cdot d + \beta$

Como $d = s_0 \cdot a + t_0 \cdot b$, tenemos que $\alpha \cdot d = \alpha \cdot s_0 \cdot a + \alpha \cdot t_0 \cdot b$

► Concluimos que $a - \beta = \alpha \cdot s_0 \cdot a + \alpha \cdot t_0 \cdot b$

Entonces tenemos que $\beta = (1 - \alpha \cdot s_0) \cdot a + (-\alpha \cdot t_0) \cdot b$

Dado que $0 \leq \beta < d$ y d es el menor elemento en D , concluimos que $\beta = 0$

► Por lo tanto $d|a$

Una identidad útil

De la misma forma concluimos que $d|b$

Sea $c \in \mathbb{N}$ tal que $c|a$ y $c|b$

Se tiene entonces que $c|(s_0 \cdot a + t_0 \cdot b)$

- ▶ Por lo tanto $c|d$, lo cual implica que $c \leq d$

Concluimos que $d|a$, $d|b$ y para todo c tal que $c|a$ y $c|b$, se tiene que $c \leq d$

- ▶ Por lo tanto $d = \text{MCD}(a, b)$



Existencia de inverso modular y la Identidad de Bézout

Teorema

a tiene inverso en módulo n si y sólo si $\text{MCD}(a, n) = 1$

Demostración: ($\Rightarrow$) Suponga que b es inverso de a en módulo n

► Entonces: $a \cdot b \equiv 1 \pmod{n}$

Se deduce que $a \cdot b = \alpha \cdot n + 1$, por lo que $1 = a \cdot b - \alpha \cdot n$

Concluimos que si $c|a$ y $c|n$, entonces $c|1$

► Por lo tanto c debe ser igual a 1, de lo que concluimos que $\text{MCD}(a, n) = 1$

Existencia de inverso modular y la Identidad de Bézout

($\Leftarrow$) Suponga que $\text{MCD}(a, n) = 1$

Por la identidad de Bézout existen $s, t \in \mathbb{Z}$ tales que:

$$1 = s \cdot n + t \cdot a$$

Por lo tanto: $a \cdot t \equiv 1 \pmod{n}$

► Concluimos que a tiene inverso en módulo n



¿Cómo podemos calcular el inverso modular?

Sabemos que **MCD** es un algoritmo eficiente para calcular el máximo común divisor entre dos números.

¡Pero este algoritmo puede hacer más!

- Puede ser extendido para calcular s y t tales que
 $\text{MCD}(a, b) = s \cdot a + t \cdot b$

Vamos a usar este algoritmo para calcular *inversos modulares*

El Algoritmo Extendido de Euclides

Suponga que $a \geq b$, y defina la siguiente sucesión:

$$\begin{aligned}r_0 &= a \\r_1 &= b \\r_{i+1} &= r_{i-1} \bmod r_i \quad (i \geq 2)\end{aligned}$$

Calculamos esta sucesión hasta un número k tal que $r_k = 0$

► Tenemos que $\text{MCD}(a, b) = r_{k-1}$

El Algoritmo Extendido de Euclides

Al mismo tiempo calculamos sucesiones s_i, t_i tales que:

$$r_i = s_i \cdot a + t_i \cdot b$$

Tenemos que: $\text{MCD}(a, b) = r_{k-1} = s_{k-1} \cdot a + t_{k-1} \cdot b$

Sean:

$$\begin{array}{ll} s_0 = 1 & t_0 = 0 \\ s_1 = 0 & t_1 = 1 \end{array}$$

Se tiene que:

$$\begin{array}{ll} r_0 &= s_0 \cdot a + t_0 \cdot b \\ r_1 &= s_1 \cdot a + t_1 \cdot b \end{array}$$

El Algoritmo Extendido de Euclides

Dado que $r_{i-1} = \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot r_i + r_{i+1} \bmod r_i$, tenemos que:

$$r_{i-1} = \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot r_i + r_{i+1}$$

Por lo tanto:

$$s_{i-1} \cdot a + t_{i-1} \cdot b = \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot (s_i \cdot a + t_i \cdot b) + r_{i+1}$$

Concluimos que:

$$r_{i+1} = (s_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot s_i) \cdot a + (t_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot t_i) \cdot b$$

Definimos entonces:

$$s_{i+1} = s_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot s_i$$

$$t_{i+1} = t_{i-1} - \lfloor \frac{r_{i-1}}{r_i} \rfloor \cdot t_i$$

El Algoritmo Extendido de Euclides

Ejemplo

Vamos a usar el algoritmo para $a = 60$ y $b = 13$

Inicialmente:

$$\begin{array}{lll} r_0 = 60 & s_0 = 1 & t_0 = 0 \\ r_1 = 13 & s_1 = 0 & t_1 = 1 \end{array}$$

Entonces tenemos que:

$$\begin{aligned} r_2 &= r_0 \bmod r_1 \\ s_2 &= s_0 - \left\lfloor \frac{r_0}{r_1} \right\rfloor \cdot s_1 \\ t_2 &= t_0 - \left\lfloor \frac{r_0}{r_1} \right\rfloor \cdot t_1 \end{aligned}$$

El Algoritmo Extendido de Euclides

Example (Continuación)

Por lo tanto:

$$r_2 = 8 \qquad s_2 = 1 \qquad t_2 = -4$$

Y el proceso continua:

$r_3 = 5$	$s_3 = -1$	$t_3 = 5$
$r_4 = 3$	$s_4 = 2$	$t_4 = -9$
$r_5 = 2$	$s_5 = -3$	$t_5 = 14$
$r_6 = 1$	$s_6 = 5$	$t_6 = -23$
$r_7 = 0$	$s_7 = -13$	$t_7 = 60$

Tenemos que: $1 = 5 \cdot 60 + (-23) \cdot 13$

El Algoritmo Extendido de Euclides y el inverso modular

Dados dos números naturales a y n , con $n \geq 2$, si el inverso de a en módulo n existe el siguiente algoritmo lo retorna, y en caso contrario indica que no existe.

```
Inverso( $a, n$ )  
  if  $\text{MCD}(a, n) > 1$  then no_existe_inverso  
  else  
     $s_0 := 1$   
     $t_0 := 0$   
     $s_1 := 0$   
     $t_1 := 1$   
     $r_0 := n$   
     $r_1 := a$ 
```

El Algoritmo Extendido de Euclides y el inverso modular

```
while  $r_1 > 1$  do  
     $aux\_s := s_0 - \lfloor \frac{r_0}{r_1} \rfloor \cdot s_1$   
     $s_0 := s_1$   
     $s_1 := aux\_s$   
     $aux\_t := t_0 - \lfloor \frac{r_0}{r_1} \rfloor \cdot t_1$   
     $t_0 := t_1$   
     $t_1 := aux\_t$   
     $r_0 := r_1$   
     $r_1 := s_1 \cdot n + t_1 \cdot a$   
return  $t_1$ 
```

Un problema fundamental: verificación de primalidad

Vamos a ver un algoritmo aleatorizado para verificar si un número es primo.

- ▶ Este algoritmo es mucho más eficiente que los algoritmos sin componentes aleatorias para este problema

El ingrediente fundamental para el algoritmo es el uso de aritmética modular.

Un primer ingrediente

Teorema (Fermat)

Sea p un número primo. Si $a \in \{0, \dots, p-1\}$, entonces $a^p \equiv a \pmod{p}$

Demostración: Por inducción en a

Para $a = 0$ y $a = 1$ se cumple trivialmente. Suponga que $a^p \equiv a \pmod{p}$ y $2 \leq (a+1) < p$

Sabemos que:

$$(a+1)^p = \sum_{k=0}^p \binom{p}{k} a^k$$

Un primer ingrediente

Por lo tanto:

$$(a+1)^p - (a+1) = (a^p - a) + \sum_{k=1}^{p-1} \binom{p}{k} a^k$$

Lema

Si $k \in \{1, \dots, p-1\}$, entonces $p \mid \binom{p}{k}$

Demostración: Sabemos que:

$$\binom{p}{k} = \frac{p!}{k! \cdot (p-k)!} = \frac{p \cdot (p-1) \cdot \dots \cdot (p-k+1)}{k!}$$

Como $k \in \{1, \dots, p-1\}$ y p es un número primo:

$\frac{(p-1) \cdot \dots \cdot (p-k+1)}{k!}$ es un número entero

Un primer ingrediente

Por lo tanto: $\binom{p}{k} = p \cdot \alpha$, donde α es un número entero

- Concluimos que $p \mid \binom{p}{k}$



Del lema concluimos que $p \mid \sum_{k=1}^{p-1} \binom{p}{k} a^k$

Por lo tanto, dado que $p \mid (a^p - a)$ por hipótesis de inducción, tenemos que: $p \mid ((a+1)^p - (a+1))$

- Concluimos que $(a+1)^p \equiv (a+1) \pmod{p}$



Un primer ingrediente

Corolario (Fermat)

Sea p un número primo. Si $a \in \{1, \dots, p-1\}$, entonces $a^{p-1} \equiv 1 \pmod{p}$

Demostración: Por teorema anterior sabemos que

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

Por lo tanto: existe un número entero α tal que

$$a^p - a = \alpha \cdot p$$

Un primer ingrediente

Dado que $a|(a^p - a)$, se tiene que $a|(\alpha \cdot p)$

Por lo tanto, dado que $a \in \{1, \dots, p-1\}$ y p es un número primo, se concluye que $a|\alpha$

Entonces: $(a^{p-1} - 1) = \frac{\alpha}{a} \cdot p$, donde $\frac{\alpha}{a}$ es un número entero.

► Concluimos que $a^{p-1} \equiv 1 \pmod{p}$



Test de primalidad: primera versión

El test de primalidad que vamos a estudiar está basado en estas propiedades ($n \geq 3$):

1. Si n es primo y $a \in \{1, \dots, n-1\}$, entonces $a^{n-1} \equiv 1 \pmod{n}$
2. Si n es compuesto, entonces existe $a \in \{1, \dots, n-1\}$ tal que $a^{n-1} \not\equiv 1 \pmod{n}$

Demostración de 2. Sea $a \in \{1, \dots, n-1\}$ tal que $\text{MCD}(a, n) > 1$

- a no tiene inverso en módulo n

Concluimos que $a^{n-1} \not\equiv 1 \pmod{n}$

- Dado que $a^{n-1} \not\equiv 1 \pmod{n}$ no puede ser inverso de a en módulo n



Test de primalidad: primera versión

Para $n \geq 2$, sea:

$$\mathbb{Z}_n^* = \{a \in \{1, \dots, n-1\} \mid \text{MCD}(a, n) = 1\}$$

Sabemos que para n compuesto: Si $a \in (\{1, \dots, n-1\} \setminus \mathbb{Z}_n^*)$, entonces $a^{n-1} \not\equiv 1 \pmod{n}$

Test de primalidad depende de cuan grande es $\mathbb{Z}_n^*$

► Función de Euler: $\phi(1) = 0$ y $\phi(n) = |\mathbb{Z}_n^*|$ para $n \geq 2$

Una cota inferior para la función ϕ de Euler

Teorema

$$\phi(n) \in \Omega\left(\frac{n}{\log_2(\log_2(n))}\right)$$

Conclusión

Para un número compuesto n , el conjunto $\mathbb{Z}_n^*$ puede tener un gran número de elementos.

- No podemos basar nuestro test en los elementos del conjunto $(\{1, \dots, n-1\} \setminus \mathbb{Z}_n^*)$

Test de primalidad: segunda versión

Una observación importante: si n es compuesto, entonces puede existir $a \in \mathbb{Z}_n^*$ tal que $a^{n-1} \not\equiv 1 \pmod n$.

► Por ejemplo: $3^{15} \bmod 16 = 11$

En lugar de considerar $\mathbb{Z}_n^*$ en el test de primalidad, consideramos:

$$J_n = \{a \in \mathbb{Z}_n^* \mid a^{n-1} \equiv 1 \pmod n\}$$

Si demostramos que para cada número compuesto n se tiene que $|J_n| \leq \frac{1}{2} \cdot |\mathbb{Z}_n^*|$, entonces tenemos un test de primalidad.

► Puesto que para p primo: $|J_p| = |\mathbb{Z}_p^*| = p - 1$

Test de primalidad: segunda versión

¿Cómo funcionaría el test de primalidad?

- ▶ ¿Con que salidas se podría equivocar? ¿Cuál sería la probabilidad de error?

¿Qué enfoque podríamos usar para demostrar que $|J_n| \leq \frac{1}{2} \cdot |\mathbb{Z}_n^*|$ (para un número n compuesto)?

- ▶ Teoría de grupos juega también un papel fundamental en el desarrollo del test de primalidad

Definición

Un conjunto G y una función (total) $\circ : G \times G \rightarrow G$ forman un grupo si:

1. Para cada $a, b, c \in G$: $(a \circ b) \circ c = a \circ (b \circ c)$
2. Existe $e \in G$ tal que para cada $a \in G$: $a \circ e = e \circ a = a$
3. Para cada $a \in G$, existe $b \in G$: $a \circ b = b \circ a = e$

Propiedades básicas

- ▶ Neutro es único: Si e_1 y e_2 satisfacen 2, entonces $e_1 = e_2$
- ▶ Inverso de cada elemento a es único: Si $a \circ b = b \circ a = e$ y $a \circ c = c \circ a = e$, entonces $b = c$

Teoría de grupos: algunos ejemplos

Ejercicios

Muestre que los siguientes son grupos:

1. $(\mathbb{Z}_n, +)$, donde $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$ y $+$ es la suma en módulo n
2. $(\mathbb{Z}_n^*, \cdot)$, donde $\cdot$ es la multiplicación en módulo n
3. $(J_n, \cdot)$, donde $\cdot$ es la multiplicación en módulo n

Teoría de grupos: subgrupos

Definition

$(H, \circ)$ es un subgrupo de un grupo $(G, \circ)$, para $\emptyset \subsetneq H \subseteq G$, si $(H, \circ)$ es un grupo.

Ejercicio

Demuestre que $(J_n, \cdot)$ es un subgrupo de $(\mathbb{Z}_n^*, \cdot)$

Propiedades básicas

- ▶ Si e_1 es el neutro en $(G, \circ)$ y e_2 es el neutro de $(H, \circ)$, entonces $e_1 = e_2$
- ▶ Para cada $a \in H$, si b es el inverso de a en $(G, \circ)$ y c es el inverso de a en $(H, \circ)$, entonces $c = b$

Subgrupos: demostración de las propiedades básicas

- ▶ Suponga que e_1 es el neutro en $(G, \circ)$ y e_2 es el neutro de $(H, \circ)$
 - ▶ Vamos a demostrar que $e_1 = e_2$

Dado que e_1 es el neutro en $(G, \circ)$: $e_2 \circ e_1 = e_2$

Dado que e_2 es el neutro en $(H, \circ)$: $e_2 \circ e_2 = e_2$

- ▶ Nótese que usamos la misma operación en H y G

Concluimos que $e_2 \circ e_1 = e_2 \circ e_2$

Subgrupos: demostración de las propiedades básicas

Sea e_2^{-1} el inverso de e_2 en $(G, \circ)$. Entonces:

$$\begin{aligned}e_2 \circ e_1 = e_2 \circ e_2 &\Rightarrow e_2^{-1} \circ (e_2 \circ e_1) = e_2^{-1} \circ (e_2 \circ e_2) \\&\Rightarrow (e_2^{-1} \circ e_2) \circ e_1 = (e_2^{-1} \circ e_2) \circ e_2 \\&\Rightarrow e_1 \circ e_1 = e_1 \circ e_2 \\&\Rightarrow e_1 = e_2\end{aligned}$$

Por lo tanto: $e_1 = e_2$

Subgrupos: demostración de las propiedades básicas

- ▶ Sea $a \in H$, y suponga que b es el inverso de a en $(G, \circ)$ y c es el inverso de a en $(H, \circ)$
 - ▶ Vamos a demostrar que $c = b$

Sea e el neutro en $(G, \circ)$ y $(H, \circ)$

- ▶ Por la primera propiedad, sabemos que $(G, \circ)$ y $(H, \circ)$ tienen el mismo neutro

Dado que b es el inverso de a en $(G, \circ)$: $a \circ b = e$

Dado que c es el inverso de a en $(H, \circ)$: $a \circ c = e$

Subgrupos: demostración de las propiedades básicas

Concluimos que: $a \circ b = a \circ c$

Sea a^{-1} el inverso de a en $(G, \circ)$. Entonces:

$$\begin{aligned}a \circ b = a \circ c &\Rightarrow a^{-1} \circ (a \circ b) = a^{-1} \circ (a \circ c) \\&\Rightarrow (a^{-1} \circ a) \circ b = (a^{-1} \circ a) \circ c \\&\Rightarrow e \circ b = e \circ c \\&\Rightarrow b = c\end{aligned}$$

Por lo tanto: $b = c$



Teoría de grupos: una propiedad fundamental

Teorema (Lagrange)

Si $(G, \circ)$ es un grupo finito y $(H, \circ)$ es un subgrupo de $(G, \circ)$, entonces $|H|$ divide a $|G|$

Demostración: Suponga que e es el elemento neutro de $(G, \circ)$ y a^{-1} es el inverso de a en $(G, \circ)$

Sea $\sim$ una relación binaria sobre G definida como:

$$a \sim b \text{ si y sólo si } b \circ a^{-1} \in H$$

Lema

$\sim$ es una relación de equivalencia.

Teorema de Lagrange: demostración del primer lema

- ▶ $a \sim a$ ya que $a \circ a^{-1} = e$ y $e \in H$
- ▶ Suponga que $a \sim b$
 - ▶ Tenemos que demostrar que $b \sim a$

Dado que $a \sim b$: $b \circ a^{-1} \in H$

- ▶ Tenemos que demostrar que $a \circ b^{-1} \in H$

Tenemos que:

$$\begin{aligned}(b \circ a^{-1}) \circ (a \circ b^{-1}) &= (b \circ (a^{-1} \circ a)) \circ b^{-1} \\ &= (b \circ e) \circ b^{-1} \\ &= b \circ b^{-1} \\ &= e\end{aligned}$$

Teorema de Lagrange: demostración del primer lema

De la misma forma concluimos que $(a \circ b^{-1}) \circ (b \circ a^{-1}) = e$

- ▶ Por lo tanto: $(b \circ a^{-1})^{-1} = a \circ b^{-1}$

Concluimos que $a \circ b^{-1}$ está en H , ya que $(H, \circ)$ es un subgrupo de $(G, \circ)$

- ▶ Suponga que $a \sim b$ y $b \sim c$
 - ▶ Tenemos que demostrar que $a \sim c$

Por hipótesis: $b \circ a^{-1} \in H$ y $c \circ b^{-1} \in H$

- ▶ Tenemos que demostrar que $c \circ a^{-1} \in H$

Pero $(c \circ b^{-1}) \circ (b \circ a^{-1}) = c \circ a^{-1}$ y $\circ$ es cerrada en H

- ▶ Por lo tanto: $c \circ a^{-1} \in H$



Teorema de Lagrange: demostración

Sea $[a]_{\sim}$ la clase de equivalencia de $a \in G$ bajo la relación $\sim$

Lema

1. $[e]_{\sim} = H$
2. Para cada $a, b \in G$: $|[a]_{\sim}| = |[b]_{\sim}|$

Del lema se concluye el teorema.

- Puesto que las clases de equivalencia de $\sim$ particionan G

Teorema de Lagrange: demostración del segundo lema

1. Se tiene que:

$$\begin{aligned} a \in [e]_{\sim} &\Leftrightarrow e \sim a \\ &\Leftrightarrow a \circ e^{-1} \in H \\ &\Leftrightarrow a \circ e \in H \\ &\Leftrightarrow a \in H \end{aligned}$$

2. Sean $a, b \in G$, y defina la función f de la siguiente forma:

$$f(x) = x \circ (a^{-1} \circ b)$$

Teorema de Lagrange: demostración del segundo lema

Se tiene que:

$$\begin{aligned}x \in [a]_{\sim} &\Rightarrow a \sim x \\&\Rightarrow x \circ a^{-1} \in H \\&\Rightarrow (x \circ a^{-1}) \circ e \in H \\&\Rightarrow (x \circ a^{-1}) \circ (b \circ b^{-1}) \in H \\&\Rightarrow (x \circ (a^{-1} \circ b)) \circ b^{-1} \in H \\&\Rightarrow f(x) \circ b^{-1} \in H \\&\Rightarrow b \sim f(x) \\&\Rightarrow f(x) \in [b]_{\sim}\end{aligned}$$

Por lo tanto: $f : [a]_{\sim} \rightarrow [b]_{\sim}$

- Vamos a demostrar que f es una biyección, de lo cual concluimos que $|[a]_{\sim}| = |[b]_{\sim}|$

Teorema de Lagrange: demostración del segundo lema

f es 1-1:

$$\begin{aligned}f(x) = f(y) &\Rightarrow x \circ (a^{-1} \circ b) = y \circ (a^{-1} \circ b) \\&\Rightarrow (x \circ (a^{-1} \circ b)) \circ (b^{-1} \circ a) = \\&\quad (y \circ (a^{-1} \circ b)) \circ (b^{-1} \circ a) \\&\Rightarrow x \circ (a^{-1} \circ (b \circ b^{-1}) \circ a) = \\&\quad y \circ (a^{-1} \circ (b \circ b^{-1}) \circ a) \\&\Rightarrow x \circ ((a^{-1} \circ e) \circ a) = y \circ ((a^{-1} \circ e) \circ a) \\&\Rightarrow x \circ (a^{-1} \circ a) = y \circ (a^{-1} \circ a) \\&\Rightarrow x \circ e = y \circ e \\&\Rightarrow x = y\end{aligned}$$

Teorema de Lagrange: demostración del segundo lema

f es sobre:

$$\begin{aligned}y \in [b]_{\sim} &\Rightarrow b \sim y \\&\Rightarrow y \circ b^{-1} \in H \\&\Rightarrow (y \circ b^{-1}) \circ (a \circ a^{-1}) \in H \\&\Rightarrow ((y \circ b^{-1}) \circ a) \circ a^{-1} \in H \\&\Rightarrow a \sim ((y \circ b^{-1}) \circ a) \\&\Rightarrow ((y \circ b^{-1}) \circ a) \in [a]_{\sim}\end{aligned}$$

Sea $x = ((y \circ b^{-1}) \circ a)$. Tenemos que:

$$\begin{aligned}f(x) &= x \circ (a^{-1} \circ b) \\&= ((y \circ b^{-1}) \circ a) \circ (a^{-1} \circ b) \\&= y \circ (b^{-1} \circ (a \circ a^{-1}) \circ b) \\&= y \circ ((b^{-1} \circ e) \circ b) \\&= y \circ (b^{-1} \circ b) \\&= y \circ e \\&= y\end{aligned}$$



Test de primalidad: segunda versión (continuación)

Pregunta pendiente: ¿Qué enfoque podríamos usar para demostrar que $|J_n| \leq \frac{1}{2} \cdot |\mathbb{Z}_n^*|$?

► ¡Usamos el Teorema de Lagrange!

Dado que $(J_n, \cdot)$ es un subgrupo de $(\mathbb{Z}_n^*, \cdot)$:

Si existe $a \in (\mathbb{Z}_n^* \setminus J_n)$, entonces $|J_n| \leq \frac{1}{2} \cdot |\mathbb{Z}_n^*|$

¿Tenemos entonces nuestro test de primalidad?

- Lamentablemente no todavía: números de Carmichael
- Pero lo que hemos aprendido va a ser fundamental para desarrollar el test de primalidad

Test de primalidad: segunda versión (continuación)

Definition

Un número n es de Carmichael si $n \geq 2$, n es compuesto y $|J_n| = |\mathbb{Z}_n^*|$

Ejemplo

561, 1105 y 1729 son números de Carmichael.

Teorema (Alford-Granville-Pomerance)

Existe un número infinito de números de Carmichael.

Test de primalidad: tercera version

Conclusión: el test basado en J_n no va a funcionar.

¿Qué hacemos entonces?

- En lugar de utilizar J_n , vamos a usar las herramientas que desarrollamos sobre el siguiente conjunto (n impar):

$$S_n = \{a \in \mathbb{Z}_n^* \mid a^{\frac{n-1}{2}} \equiv 1 \pmod{n} \text{ ó } a^{\frac{n-1}{2}} \equiv -1 \pmod{n}\}$$

¡Esto sí funciona!

Test de primalidad: un intento exitoso

Vamos a diseñar un test de primalidad considerando los conjuntos:

$$\begin{aligned}S_n^+ &= \{a \in \mathbb{Z}_n^* \mid a^{\frac{n-1}{2}} \equiv 1 \pmod{n}\} \\S_n^- &= \{a \in \mathbb{Z}_n^* \mid a^{\frac{n-1}{2}} \equiv -1 \pmod{n}\} \\S_n &= S_n^+ \cup S_n^-\end{aligned}$$

Para hacer esto necesitamos estudiar algunas propiedades de los conjuntos S_n^+ , S_n^- y S_n

- Consideramos primero el caso en que n es primo, y luego el caso en que n es compuesto

Una propiedad fundamental de S_n para n primo

Proposición

Si $n \geq 3$ es primo, entonces $S_n = \mathbb{Z}_n^$*

Demostración: Si $a \in \{1, \dots, n-1\}$, tenemos que $a^{n-1} \equiv 1 \pmod n$

Por lo tanto $(a^{\frac{n-1}{2}})^2 \equiv 1 \pmod n$, de lo cual se deduce que:

$$(a^{\frac{n-1}{2}} + 1) \cdot (a^{\frac{n-1}{2}} - 1) \equiv 0 \pmod n$$

Así, dado que n es primo se concluye que $a^{\frac{n-1}{2}} \equiv 1 \pmod n$ ó $a^{\frac{n-1}{2}} \equiv -1 \pmod n$

► ¿Por qué?



La estructura de S_n y las raíces modulares

Definition

b es una raíz cuadrada de a en módulo n si $b^2 \equiv a \pmod{n}$

Ejemplo

3 es una raíz cuadrada de 9 en módulo 16, y 3 es una raíz cuadrada de 4 en módulo 5.

Vamos a ver que hay una relación estrecha entre S_n y las raíces cuadradas modulares.

- Esta relación es fundamental para el test de primalidad

La estructura de S_n y las raíces modulares

Teorema

Sea $n \geq 3$ un primo y $a \in \{1, \dots, n-1\}$. Entonces a tiene raíz cuadrada en módulo n si y sólo si $a^{\frac{n-1}{2}} \equiv 1 \pmod{n}$

Demostración: ($\Rightarrow$) Suponga que a tiene raíz cuadrada en módulo n

► Sea $b \in \{1, \dots, n-1\}$ tal que $b^2 \equiv a \pmod{n}$

Se tiene que:

$$\begin{aligned} a^{\frac{n-1}{2}} &\equiv (b^2)^{\frac{n-1}{2}} \pmod{n} \\ &\equiv b^{n-1} \pmod{n} \\ &\equiv 1 \pmod{n} \end{aligned}$$

Existencia de raíces modulares: demostración

($\Leftarrow$) Para demostrar esta dirección del teorema, usamos el siguiente lema.

Sea $p(x)$ el polinomio:

$$p(x) = \sum_{i=0}^k a_i x^i,$$

donde $k \geq 1$, $a_k \in \{1, \dots, n-1\}$ y cada $a_j \in \{0, \dots, n-1\}$
($0 \leq j \leq k-1$)

Decimos que a es una raíz de $p(x)$ en módulo n si $p(a) \equiv 0 \pmod{n}$

Existencia de raíces modulares: demostración

Lema

$p(x)$ tiene a lo más k raíces en módulo n

Demostración: Decimos que dos polinomios $p_1(x)$ y $p_2(x)$ son congruentes en módulo n si para todo $a \in \{0, \dots, n-1\}$:

$$p_1(a) \equiv p_2(a) \pmod{n}$$

Notación

$$p_1(x) \equiv p_2(x) \pmod{n}$$

Sea a una raíz de $p(x)$ en módulo n

Existencia de raíces modulares: demostración

Vamos a demostrar que existe un polinomio $q(x)$ de grado $k - 1$ tal que:

$$p(x) \equiv (x - a) \cdot q(x) \pmod{n}$$

Pero antes de demostrar esto, vamos a mostrar que de esta propiedad se concluye que el lema es cierto.

Si c es una raíz de $p(x)$ en módulo n , entonces $p(c) \equiv 0 \pmod{n}$

- ▶ Como $p(x) \equiv (x - a) \cdot q(x) \pmod{n}$, concluimos que
 $(c - a) \cdot q(c) \equiv 0 \pmod{n}$

Dado que n es primo, si $d \cdot e \equiv 0 \pmod{n}$, entonces $d \equiv 0 \pmod{n}$ o $e \equiv 0 \pmod{n}$

- ▶ Tenemos entonces que $c \equiv a \pmod{n}$ o $q(c) \equiv 0 \pmod{n}$

Existencia de raíces modulares: demostración

Así, tenemos que c es la raíz a que ya habíamos identificado o es una raíz de $q(x)$ en módulo n

Concluimos que el número de raíces de $p(x)$ en módulo n es menor o igual a uno más el número de raíces de $q(x)$ en módulo n

- ▶ Como $q(x)$ tiene grado $k - 1$, si continuamos usando este argumento (o usamos inducción) concluimos que el número de raíces de $p(x)$ es menor o igual a k

Existencia de raíces modulares: demostración

Nótese que el argumento anterior no funciona si n es compuesto.

- ▶ Dado que podemos tener d y e tales que $d \cdot e \equiv 0 \pmod{n}$, $d \not\equiv 0 \pmod{n}$ y $e \not\equiv 0 \pmod{n}$

De hecho, si n es compuesto no es necesariamente cierto que el número de raíces de un polinomio está acotado superiormente por su grado.

Ejemplo

Si $n = 35$, tenemos que $5 \cdot 7 \equiv 0 \pmod{35}$, pero $5 \not\equiv 0 \pmod{35}$ y $7 \not\equiv 0 \pmod{35}$

En este caso tenemos cuatro raíces para el polinomio $p(x) = x^2 - 1$

- ▶ Ya que $1^2 \equiv 1 \pmod{35}$, $6^2 \equiv 1 \pmod{35}$, $29^2 \equiv 1 \pmod{35}$ y $34^2 \equiv 1 \pmod{35}$

Existencia de raíces modulares: demostración

Volvemos entonces a la demostración de que existe un polinomio $q(x)$ de grado $k - 1$ tal que:

$$p(x) \equiv (x - a) \cdot q(x) \pmod{n}$$

Definimos $q(x)$ como:

$$q(x) = \sum_{i=0}^{k-1} b_i x^i,$$

donde $b_i = a_{i+1} + a_{i+2} \cdot a + \cdots + a_k \cdot a^{k-1-i}$

Existencia de raíces modulares: demostración

Se tiene que:

$$\begin{aligned}(x - a) \cdot q(x) &= \left(\sum_{i=0}^{k-1} b_i x^{i+1} \right) + \left(\sum_{i=0}^{k-1} (-a \cdot b_i) x^i \right) \\&= \left(\sum_{i=1}^k b_{i-1} x^i \right) + \left(\sum_{i=0}^{k-1} (-a \cdot b_i) x^i \right) \\&= b_{k-1} \cdot x^k + \left(\sum_{i=1}^{k-1} (b_{i-1} - a \cdot b_i) x^i \right) - a \cdot b_0\end{aligned}$$

Así, dado que:

$$b_{k-1} = a_k$$

Existencia de raíces modulares: demostración

Y dado que para $i \in \{1, \dots, k-1\}$:

$$\begin{aligned}(b_{i-1} - a \cdot b_i) &= a_i + a_{i+1} \cdot a + \dots + a_k \cdot a^{k-i} - \\ &\quad a \cdot (a_{i+1} + a_{i+2} \cdot a + \dots + a_k \cdot a^{k-1-i}) \\ &= a_i + a_{i+1} \cdot a + \dots + a_k \cdot a^{k-i} - \\ &\quad a_{i+1} \cdot a - a_{i+2} \cdot a^2 - \dots - a_k \cdot a^{k-1} \\ &= a_i\end{aligned}$$

Concluimos que:

$$(x - a) \cdot q(x) = \left(\sum_{i=1}^k a_i \cdot x^i \right) - a \cdot b_0$$

Existencia de raíces modulares: demostración

Pero:

$$\begin{aligned} -a \cdot b_0 &= -a \cdot (a_1 + a_2 \cdot a + \cdots + a_k \cdot a^{k-1}) \\ &= -a_1 \cdot a - a_2 \cdot a^2 - \cdots - a_k \cdot a^k \end{aligned}$$

De lo cual deducimos que:

$$a_0 \equiv -a \cdot b_0 \pmod{n},$$

ya que $a_k \cdot a^k + \cdots + a_1 \cdot a + a_0 \equiv 0 \pmod{n}$

Tenemos entonces que:

$$(x - a) \cdot q(x) \equiv p(x) \pmod{n}$$



Existencia de raíces modulares: demostración

Volvemos a la demostración inicial:

- ▶ Queremos demostrar que si $a^{\frac{n-1}{2}} \equiv 1 \pmod{n}$, entonces a tiene raíz en módulo n

$$\text{Sea } R = \{b^2 \mid 1 \leq b \leq \frac{n-1}{2}\}$$

Como $b^2 \equiv (n-b)^2 \pmod{n}$, se tiene que:

a tiene raíz en módulo n si y sólo si $a \in R$

Existencia de raíces modulares: demostración

Por ($\Rightarrow$) sabemos que:

$$R \subseteq \{a \in \{1, \dots, n-1\} \mid a^{\frac{n-1}{2}} \equiv 1 \pmod{n}\}$$

Además, sabemos que si $1 \leq b < c \leq \frac{n-1}{2}$ y $b^2 \equiv c^2 \pmod{n}$:

$$(c - b) \cdot (c + b) \equiv 0 \pmod{n}$$

Así, dado que $2 \leq b + c \leq n - 1$, concluimos que $b \equiv c \pmod{n}$

► Dado que n es primo

Obtenemos una contradicción puesto que $1 \leq (c - b) \leq \frac{n-1}{2}$

► Por lo tanto: $|R| = \frac{n-1}{2}$

Existencia de raíces modulares: demostración

Sea $p(x) = x^{\frac{n-1}{2}} - 1$. También sabemos que $p(x)$ tiene a lo más $\frac{n-1}{2}$ raíces en módulo n .

► Por lo tanto: $|\{a \in \{1, \dots, n-1\} \mid a^{\frac{n-1}{2}} \equiv 1 \pmod{n}\}| \leq \frac{n-1}{2}$

Concluimos que:

$$\frac{n-1}{2} = |R| \leq |\{a \in \{1, \dots, n-1\} \mid a^{\frac{n-1}{2}} \equiv 1 \pmod{n}\}| \leq \frac{n-1}{2}$$

Por lo tanto:

$$R = \{a \in \{1, \dots, n-1\} \mid a^{\frac{n-1}{2}} \equiv 1 \pmod{n}\}$$

Así, si $a^{\frac{n-1}{2}} \equiv 1 \pmod{n}$, se tiene que a tiene raíz en módulo n



Una propiedad fundamental de S_n^+ y S_n^- para n primo

Si $n \geq 3$ es primo, dado que $S_n = \mathbb{Z}_n^*$, obtenemos por el teorema:

- Si a no tiene raíz en módulo n , entonces $a^{\frac{n-1}{2}} \equiv -1 \pmod{n}$

Así, obtenemos el siguiente corolario de los resultados anteriores.

Corolario

Si $n \geq 3$ es primo:

$$|S_n^+| = |S_n^-| = \frac{n-1}{2}$$

Una propiedad fundamental de S_n para n compuesto

Teorema

Sea $n = n_1 \cdot n_2$, donde $n_1, n_2 \geq 3$ y $\text{MCD}(n_1, n_2) = 1$. Si existe $a \in \mathbb{Z}_n^*$ tal que $a^{\frac{n-1}{2}} \equiv -1 \pmod{n}$, entonces:

$$|S_n| \leq \frac{1}{2} \cdot |\mathbb{Z}_n^*|$$

Para demostrar el teorema necesitamos el Teorema Chino del resto

Para recordar: un teorema muy útil

Teorema (Chino del Resto)

Suponga que $\text{MCD}(m, n) = 1$. Para todo a y b , existe c tal que:

$$c \equiv a \pmod{m}$$

$$c \equiv b \pmod{n}$$

Demostración: Dado que $\text{MCD}(m, n) = 1$, existen d y e tales que:

$$n \cdot d \equiv 1 \pmod{m}$$

$$m \cdot e \equiv 1 \pmod{n}$$

Sea $c = a \cdot n \cdot d + b \cdot m \cdot e$

Se tiene que:

$$c \equiv a \pmod{m}$$

$$c \equiv b \pmod{n}$$



La demostración del teorema inicial

Suponga que $a \in \mathbb{Z}_n^*$ y $a^{\frac{n-1}{2}} \equiv -1 \pmod{n}$

Por Teorema Chino del Resto, existe b tal que:

$$b \equiv a \pmod{n_1}$$

$$b \equiv 1 \pmod{n_2}$$

Entonces: $a = \alpha \cdot n_1 + b$ y $1 = \beta \cdot n_2 + b$

- Por lo tanto $\text{MCD}(b, n) = 1$, ya que $n = n_1 \cdot n_2$ y $a \in \mathbb{Z}_n^*$

La demostración del teorema inicial

Además, tenemos que:

$$\begin{array}{rclcl} b^{\frac{n-1}{2}} & \equiv & a^{\frac{n-1}{2}} & \equiv & -1 \pmod{n_1} \\ & & b^{\frac{n-1}{2}} & \equiv & 1 \pmod{n_2} \end{array}$$

Dado que $n = n_1 \cdot n_2$ concluimos que:

$$\begin{array}{rcl} b^{\frac{n-1}{2}} & \not\equiv & 1 \pmod{n} \\ b^{\frac{n-1}{2}} & \not\equiv & -1 \pmod{n} \end{array}$$

La demostración del teorema inicial

Sea $c = (b \bmod n)$. Concluimos que $c \notin S_n$ y $c \in \mathbb{Z}_n^*$

► Por lo tanto: $S_n \subsetneq \mathbb{Z}_n^*$

Pero se tiene que $(S_n, \cdot)$ es un subgrupo de $(\mathbb{Z}_n^*, \cdot)$

► ¿Por qué?

Por Teorema de Lagrange: $|S_n| \leq \frac{1}{2} \cdot |\mathbb{Z}_n^*|$



Un test de primalidad aleatorizado

Ya tenemos los ingredientes esenciales para el test de primalidad

- ▶ Sólo nos falta implementar algunas funciones auxiliares

Necesitamos desarrollar un algoritmo eficiente para determinar si un número n es la potencia (no trivial) de otro número.

Verificando si un número es la potencia de otro

Primero necesitamos una función para calcular n^k

- Usamos el algoritmo de exponenciación rápida pero sin considerar el módulo

```
EXP( $n, k$ )  
  if  $k = 1$  then return  $n$   
  else if  $k$  es par then  
     $val := \mathbf{EXP}(n, \frac{k}{2})$   
    return  $val \cdot val$   
  else  
     $val := \mathbf{EXP}(n, \frac{k-1}{2})$   
    return  $val \cdot val \cdot n$ 
```

Verificando si un número es la potencia de otro

Dado un número natural $n \geq 2$, la siguiente función verifica si existen $m, k \in \mathbb{N}$ tales que $k \geq 2$ y $n = m^k$

EsPotencia(n)

if $n \leq 3$ **then return** no

else

for $k := 2$ **to** $\lfloor \log_2(n) \rfloor$ **do**

if **TieneRaízEntera**($n, k, 1, n$) **then return** sí

return no

Verificando si un número es la potencia de otro

La siguiente función verifica si existe $m \in \{i, \dots, j\}$ tal que $n = m^k$

- Vale decir, la llamada **TieneRaízEntera**(n, k, i, j) verifica si n tiene raíz k -ésima entera

TieneRaízEntera(n, k, i, j)

 if $i = j$ then

 if **EXP**(i, k) = n then return sí

 else return no

 else if $i < j$ then

$p := \lfloor \frac{i+j}{2} \rfloor$

$val := \mathbf{EXP}(p, k)$

 if $val = n$ then return sí

 else if $val < n$ then return **TieneRaízEntera**($n, k, p + 1, j$)

 else return **TieneRaízEntera**($n, k, i, p - 1$)

 else return no

La complejidad de **EsPotencia**

Consideramos la multiplicación de números enteros como la operación básica a contar

Tenemos que:

- ▶ En el peor caso **EsPotencia**(n) realiza $(\lfloor \log_2(n) \rfloor - 1)$ llamadas a la función **TieneRaízEntera**
- ▶ Existe $c \in \mathbb{N}$ tal que la llamada **TieneRaízEntera**($n, k, 1, n$) realiza en el peor caso a lo más $c \cdot \log_2(n)$ llamadas a la función **EXP**
- ▶ **EXP**(n, k) en el peor caso es $O(\log_2(k))$

La complejidad de **EsPotencia**

Concluimos que **EsPotencia**(n) en el peor caso es $O([\log_2(n)]^3)$

Vale decir, **EsPotencia** en el peor caso es de orden polinomial en el tamaño de la entrada

Un test de primalidad aleatorizado

El siguiente algoritmo aleatorizado determina si un número entero $n \geq 2$ es primo.

- ▶ Si n es primo el algoritmo retorna PRIMO, en caso contrario retorna COMPUESTO

El algoritmo también recibe como entrada un valor entero $k \geq 1$ que es usado para controlar la probabilidad de error

Un test de primalidad aleatorizado

TestPrimalidad(n, k)

if $n = 2$ then return PRIMO

else if n es par then return COMPUESTO

else if **EsPotencia**(n) then return COMPUESTO

else

 sea $a_1, \dots, a_k$ una secuencia de números elegidos de
 manera uniforme e independiente desde $\{1, \dots, n-1\}$

for $i := 1$ to k do

 if **MCD**(a_i, n) > 1 then return COMPUESTO

 else $b_i := \text{EXP}(a_i, \frac{n-1}{2}, n)$

neg := 0

for $i := 1$ to k do

 if $b_i \equiv -1 \pmod n$ then neg := neg + 1

 else if $b_i \not\equiv 1 \pmod n$ then return COMPUESTO

if neg = 0 then return COMPUESTO

else return PRIMO

Test de primalidad: probabilidad de error

TestPrimalidad se puede equivocar de dos formas:

- ▶ Suponga que $n \geq 3$ es primo. En este caso **TestPrimalidad** da una respuesta incorrecta si $b_i \equiv 1 \pmod{n}$ para todo $i \in \{1, \dots, k\}$

Dado que $|S_n^+| = |S_n^-| = \frac{n-1}{2}$:

- ▶ La probabilidad de que para un número a elegido con distribución uniforme desde $\{1, \dots, n-1\}$ se tenga que $a^{\frac{n-1}{2}} \equiv 1 \pmod{n}$ es $\frac{1}{2}$

Por lo tanto, la probabilidad de que **TestPrimalidad** diga COMPUESTO para $n \geq 3$ primo es $\frac{1}{2^k}$

Test de primalidad: probabilidad de error

- ▶ Suponga que n es compuesto, n es impar y n no es de la forma m^ℓ con $\ell \geq 2$
 - ▶ Si n es par o n es de la forma m^ℓ con $\ell \geq 2$, entonces **TestPrimalidad** da la respuesta correcta COMPUESTO

Tenemos entonces que $n = n_1 \cdot n_2$ con $n_1 \geq 3$, $n_2 \geq 3$ y $\text{MCD}(n_1, n_2) = 1$

Además debe existir $a \in \{1, \dots, n-1\}$ tal que $\text{MCD}(a, n) = 1$
y $a^{\frac{n-1}{2}} \equiv -1 \pmod{n}$

- ▶ Si esto no es cierto **TestPrimalidad** retorna COMPUESTO, dado que si **TestPrimalidad** logra llegar a la última instrucción **if** entonces *neg* necesariamente es igual a 0

Test de primalidad: probabilidad de error

Concluimos que $|S_n| \leq \frac{1}{2} \cdot |\mathbb{Z}_n^*|$

- Por la caracterización que dimos de S_n para n compuesto

Vamos a utilizar este resultado para acotar la probabilidad de error:

$$\Pr\left(\left(\bigwedge_{i=1}^k \text{MCD}(a_i, n) = 1 \wedge (b_i \equiv 1 \pmod{n} \vee b_i \equiv -1 \pmod{n})\right) \wedge \left(\bigvee_{j=1}^k b_j \equiv -1 \pmod{n}\right)\right)$$

Test de primalidad: probabilidad de error

Tenemos que:

$$\Pr\left(\left(\bigwedge_{i=1}^k \text{MCD}(a_i, n) = 1 \wedge (b_i \equiv 1 \bmod n \vee b_i \equiv -1 \bmod n)\right) \wedge \left(\bigvee_{j=1}^k b_j \equiv -1 \bmod n\right)\right) \leq$$
$$\Pr\left(\bigwedge_{i=1}^k \text{MCD}(a_i, n) = 1 \wedge (b_i \equiv 1 \bmod n \vee b_i \equiv -1 \bmod n)\right)$$

Por lo tanto sólo necesitamos una cota superior para la última expresión.

Test de primalidad: probabilidad de error

Tenemos que:

$$\begin{aligned} & \Pr\left(\bigwedge_{i=1}^k \text{MCD}(a_i, n) = 1 \wedge (b_i \equiv 1 \bmod n \vee b_i \equiv -1 \bmod n)\right) \\ &= \prod_{i=1}^k \Pr(\text{MCD}(a_i, n) = 1 \wedge (b_i \equiv 1 \bmod n \vee b_i \equiv -1 \bmod n)) \\ &= \prod_{i=1}^k \Pr((b_i \equiv 1 \bmod n \vee b_i \equiv -1 \bmod n) | \text{MCD}(a_i, n) = 1) \cdot \\ & \qquad \qquad \qquad \Pr(\text{MCD}(a_i, n) = 1) \\ &\leq \prod_{i=1}^k \Pr((b_i \equiv 1 \bmod n \vee b_i \equiv -1 \bmod n) | \text{MCD}(a_i, n) = 1) \\ &= \prod_{i=1}^k \Pr(a_i \in S_n | a_i \in \mathbb{Z}_n^*) \leq \prod_{i=1}^k \frac{1}{2} = \frac{1}{2^k} \end{aligned}$$

Test de primalidad: probabilidad de error

Concluimos que la probabilidad de que el test diga PRIMO para el valor compuesto n está acotada por $\frac{1}{2^k}$

En ambos casos (si n es primo o compuesto) la probabilidad de error del algoritmo está acotada por $\frac{1}{2^k}$

- ▶ ¡Si $k = 100$, esta probabilidad está acotada por $\frac{1}{2^{100}}$!