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 $a_1, \ldots, a_k$ una secuencia de números elegidos de
                manera uniforme e independiente desde $\{1, \ldots, n-1\}$
        for $i := 1$ **to** k **do**
            if **MCD**$(a_i, n) > 1$ **then return** COMPUESTO
            else $b_i :=$ **Exp**$(a_i, \frac{n-1}{2}, n)$
        $neg := 0$
        for $i := 1$ **to** k **do**
            if $b_i \equiv -1 \bmod n$ **then** $neg := neg + 1$
            else if $b_i \not\equiv 1 \bmod n$ **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 $b_i \equiv 1 \bmod n$ para todo $i \in \{1, \ldots, k\}$

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

  - La probabilidad de que para un número a elegido con distribución uniforme desde $\{1, \ldots, n-1\}$ se tenga que $a^{\frac{n-1}{2}} \equiv 1 \bmod n$ 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 $a \in \{1, \ldots, n-1\}$ tal que $\text{MCD}(a, n) = 1$ y $a^{\frac{n-1}{2}} \equiv -1 \bmod n$

- 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:

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

Test de primalidad: probabilidad de error

Tenemos que:

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

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

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

Test de primalidad: probabilidad de error

Tenemos que:

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

$$= \prod_{i=1}^{k} \Pr(\text{MCD}(a_i, n) = 1 \wedge (b_i \equiv 1 \bmod n \vee b_i \equiv -1 \bmod n))$$

$$= \prod_{i=1}^{k} \Pr((b_i \equiv 1 \bmod n \vee b_i \equiv -1 \bmod n) | \text{MCD}(a_i, n) = 1) \cdot$$

$$\Pr(\text{MCD}(a_i, n) = 1)$$

$$\leq \prod_{i=1}^{k} \Pr((b_i \equiv 1 \bmod n \vee b_i \equiv -1 \bmod n) | \text{MCD}(a_i, n) = 1)$$

$$= \prod_{i=1}^{k} \Pr(a_i \in S_n | a_i \in \mathbb{Z}_n^*) \leq \prod_{i=1}^{k} \frac{1}{2} = \frac{1}{2^k}$$

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 $k = 100$, está probabilidad está acotada por $\frac{1}{2^{100}}$!