Criptografía y Computación
grado en ingeniería informática

Práctica 1

Primalidad

Autor
Vladislav Nikolov Vasilev

Rama
Computación y Sistemas Inteligentes

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Curso 2019-2020

Índice

Instrucciones de ejecución

Ejercicio 1

En este ejercicio se pide implementar una función que realice el test de Miller-Rabin dados un número impar n y un testigo a tal que $2 \leq a \leq n-2$. La función debe devolver verdadero en caso de que n sea probable primo y falso en caso contrario.

Por una parte, para realizar el test de Miller-Rabin necesitamos una función que calcule la descomposición de $n-1$ como $2^u * s$, donde s es un número impar. Esta función se ha implementado de la siguiente forma:

```python
def descomposicion(n):
    # Inicializar u y s
    u = 0
    s = n

    while s % 2 == 0:
        u += 1
        s = s // 2

    return u, s
```

La función que realiza el test de Miller-Rabin para un n y un a dados es la siguiente:

```python
def miller_rabin(n, a):
    # 1. Descomponer n-1 como 2^u * s con s impar
    u, s = descomposicion(n-1)

    # 2. Calcular a = a^s mod n
    a = potencia_modular(a, s, n)

    # Si a == 1 o a == n-1, el numero es posible primo
    if a == 1 or a == n-1:
        return True

    for i in range(1, u):
        a = potencia_modular(a, 2, n)

        # Si a == 1 sin haber pasado por n-1, el numero no es primo
        # ya que tiene mas de una solucion a x^2 - 1 = 0
        if a == 1:
            return False

        """
        Si a == n-1, el siguiente valor sera 1, por lo tanto,
        cumpliria el test de Fermat y tendria solo dos soluciones a
```

```
        la ecuacion x^2 - 1 = 0. Puede ser primo
        """
        if a == n-1:
            return True

    return False
```

Se ha probado la función anterior con $n = 1729$ y con dos testigos: $a_1 = 2$ y $a_2 = 10$. En el primero caso, la función ha determinado que n no es primo, mientras que en el segundo caso ha determinado que sí que lo es. Este comportamiento es el esperado, ya que sabemos que $1729 = 7 \cdot 247$ y que por tanto no es primo, y que $a = 10$ es un falso testigo.

Ejercicio 2

En este ejercicio se ha pedido que se implemente una función que realice el test de Miller-Rabin escogiendo m testigos aleatorios. La función es la siguiente:

```
def test_primalidad(n, m):
    for i in range(m):
        # Escoger testigo tal que 2 <= a <= n-2
        a = random.randint(2, n-2)

        es_prob_primo = miller_rabin(n, a)

        if not es_prob_primo:
            return False

    return True
```

En el momento en el que el test de Miller-Rabin devuelva falso, se ha conseguido determinar que el número no es probable primo.

Ejercicio 3

Ejercicio 4

Ejercicio 5

Ejercicio 6

Ejercicio 7

Ejercicio 8