PRÁCTICA 1

ARITMÉTICA MODULAR

Esta práctica nos introduce en algunos algoritmos de aritmética modular que son necesarios para el desarrollo de la criptografía de clave pública. Se trata de conocer e implementar algunos de ellos, y de observar las diferencias del tiempo de cálculo entre distintas operaciones. A lo largo de toda la práctica se utilizarán números naturales con un número elevado de cifras, por lo que será necesario optar por un lenguaje de programación que admita dichos cálculos, o bien adaptar, mediante el uso de bibliotecas específicas, el habitual para llevar a cabo estas tareas.

1.1. Algoritmos previos.

Para el desarrollo de la práctica puede que sean necesarias algunas funciones auxiliares que deben ser implementadas. Encontraréis en Descargas de Swad unos apuntes sobre números naturales y enteros que contienen toda la información que necesitáis. Podéis consultar:

Algoritmo de Euclides.

Cálculo del inverso.

Estos algoritmos sólo son auxiliares, no son el objeto de la práctica.

1.2. Primera parte: potencia y logaritmo.

1. Implementad un programa que pida como entrada 3 números naturales de tamaño arbitrario a, b y p, y dé como salida $a^b \pmod p$.

2. Implementad un programa que pida como entrada 3 enteros de tamaño arbitrario a, b y p (ahora se usará suponiendo que es un **número primo**), y dé como salida $log_a\, b \pmod p$.

3. Para cada número primo de la siguiente lista

$$13177$$
$$251477$$
$$4461881$$
$$9843079$$
$$15444521$$
$$472251991$$
$$5463458053$$
$$735773424149$$
$$5746175430239$$

Elegid al azar dos números a, b tales que $2 \leq a$, $b \leq p-2$ y con aproximadamente el mismo número de dígitos que p y realizad las operaciones:

a) calcular $c = a^b \pmod{p}$ y medir el tiempo utilizado para el cómputo;

b) calcular $d = log_a\, c \pmod{p}$ y medir el tiempo utilizado para el cómputo;

Elaborad entonces una tabla comparativa de los tiempos de cálculo de ambas operaciones en función del número de dígitos de p.

1.3. Segunda parte: test de primalidad.

Se trata de determinar cuándo un número natural dado es primo. Para ello se utilizará el test de primalidad probabilístico llamado de Miller-Rabin. Este test aparece en los apuntes de la asignatura en las páginas 192 y siguientes. Las condiciones del programa que se pide son:

La forma básica debe tener como entrada un número impar p, y un número natural n, y comprobar si el número p pasa el test de primalidad n veces.

También debe tener la opción de elegir aleatoriamente el número impar p señalando el número de dígitos que deseamos que tenga.

La salida debe ser la respuesta "no es primo" o "es probable primo" según corresponda. Puede añadirse la probabilidad de que el núemro sea efectivamente primo (que depende del número de rondas que se pase el test).

Se valorará que como parámetro de entrada pueda darse la probabilidad con la que deseamos que sea probrada la primalidad, en lugar del número de rondas.

Os puede ser de utilidad una página donde podéis encontrar información sobre números primos:

http://primes.utm.edu/

1.4. Qué debe ser entregado.

★ El código fuente.

★ Los ficheros ejecutables (para linux).

★ Un breve manual (una o dos páginas) en el que se expliquen las instrucciones para la utilización del programa.

★ La tabla comparativa de los tiempos empleados en los cálculos de la potencia y el logaritmo que se solicita en el apartado 3 de la primera parte.

1.5. Cuándo y cómo se entrega la práctica.

La práctica deberá ser entregada durante el horario asignado a prácticas o previa cita en otro horario. Una vez entregada se subirán los ficheros a Mis trabajos de SWAD por el representante del grupo.

La fecha límite de entrega será el Martes 29 de Marzo de 2011.