Códigos y Criptografía. Grado en Ingeniería Informática (2022)-Universidad de Valladolid
Primera entrega de ejercios (10 % nota final)

Fecha de entrega: 20 de noviembre de 2022, hora: 23:59.

Se espera la resolución de los ejercicios en SageMath. Se deben entregar por email a la dirección diego.ruano@uva.es tanto la sesión de SageMath (ejecutable) como una impresión en pdf de dicha sesión. **La resolución se debe acompañar de una breve explicación**.

Ejercicio 1

- Explica brevemente como definir un cuerpo finito, como se representan sus elementos y como se opera con ellos en SageMath. Considera tanto uno con un número primo de elementos como uno con una potencia de un número primo de elementos.

Ejercicio 2

- Muestra con un ejemplo comentado como usar la clase de códigos lineales en SageMath. En particular como definir un código lineal y como trabajar/obtener su matriz generadora, control, parámetros, descodificación, polinomio de pesos, código dual, ... (los conceptos vistos en clase).

Ejercicio 3

- Implementa el algoritmo de decodificación de códigos Reed-Solomon (decodificación única) del Ejemplo 4.2.1 del libro de Justesen-Høholdt. Muestra con varias palabras recibidas las diferentes situaciones en la decodificación que pueden suceder.

- Implementa el algoritmo de decodificación de códigos Reed-Solomon (decodificación única) visto en clase (sección 4.2 del libro de Justesen-Høholdt) y calcula un ejemplo donde se descodifican varias palabras.

- Implementa el algoritmo de decodificación en lista de códigos Reed-Solomon del Ejemplo 4.3.1 del libro de Justesen-Høholdt.

- Considera el código Reed-Solomon del Ejemplo 4.3.1 del libro de Justesen-Høholdt y diferentes valores de τ (número de errores admitidos), calcula de forma experimental la probabilidad de que el algoritmo de decodificación en lista devuelva una lista con más de un elemento.

Ejercicio 4

- Implementa una función que introduzca con probabilidad p, de tu elección, un error en cada bit de una palabra.

- Para un código C, de tu elección y para una probabilidad de error en cada bit, p, de tu elección, calcula la probabilidad de que una palabra recibida sea decodificada correctamente.
- Juega con el cálculo anterior subiendo y bajando la probabilidad de error. ¿Para que valores la probabilidad de error el código corrector funciona con una alta fiabilidad?

Diego Ruano