Ejercicio AvCont-4(c)

- Enunciado – parte c:

Después de haber implementado en Java un compresor/descompresor de datos binarios mediante el algoritmo LZ-77 que cumpla las especificaciones indicadas en la parte (a) del ejercicio, analizaremos ahora la **capacidad de LZ-77 para comprimir audio.**

f) Haced las modificaciones necesarias en el programa anterior para que sea capaz de **leer ficheros de audio en formato .wav, obtener el código binario natural de cada unos de los enteros (int) de los que se compone el sonido (16bits por dato) y devolverlo como una *String binaria* procesable por vuestros métodos de compresión /descompresión en LZ-77**. Ayuda: utilizar los métodos de la clase WavReader que se os proporciona.

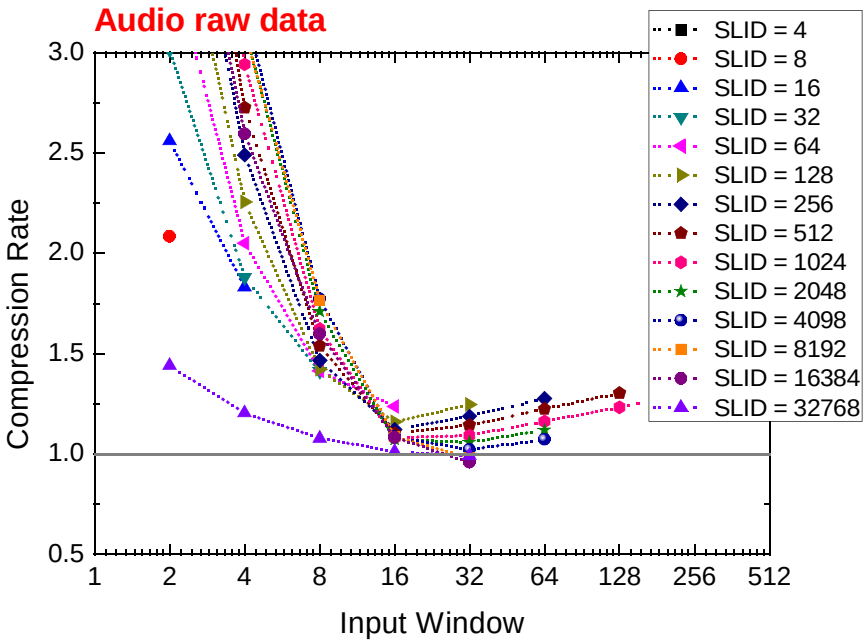
g) En el Ej. AvCont-5, implementasteis métodos para hallar el código Rice de un número entero, dado un parámetro de compresión M. Modificad el programa anterior para que, **codifique los datos enteros (int) de audio en codificación Rice antes de realizar la compresión LZ-77**. Emplead el parámetro de compresión M que juzguéis más adecuado según lo discutido en clase (es decir, con el que se logra más compresión).

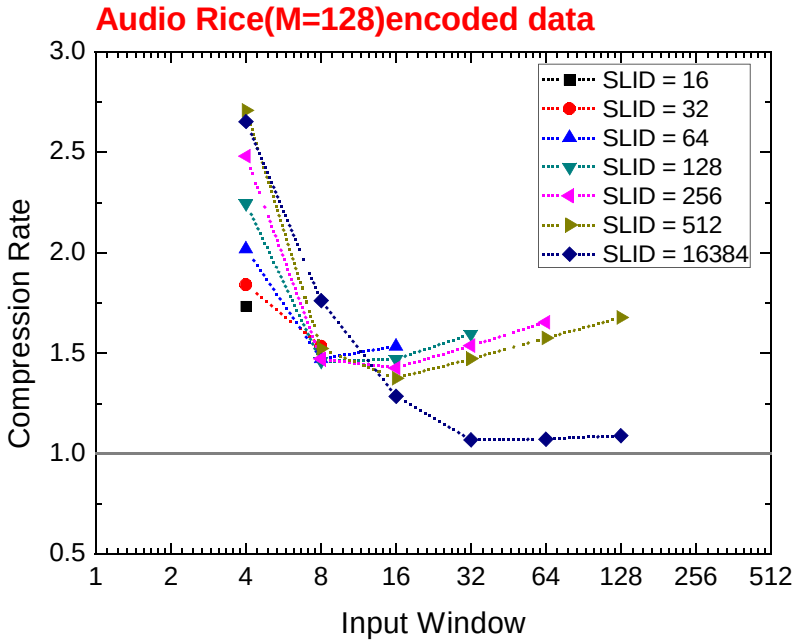
h) Utilizando los programas anteriores **comprimid en "LZ-77" i en "Rice+LZ-77" el archivo "data.wav"** que se os proporciona (sonido en formato wav monocanal). Fijando distintos valores de M_{des} y M_{ent} entre 4 y 4096 y analizad el factor de compresión en cada caso. ¿Qué conclusiones sacáis?

(cada apartado correcto cuenta como uno de los ejercicios hechos anteriormente)
(podéis trabajar por grupos)
(subir resolución -código fuente incluido- a campus virtual antes de 13 dic. 2010)

Ejercicio AvCont-4(c)

- Solución – parte c:





LZ77: dificultades para comprimir señales de audio (muy aleatorias)

(se consigue si M_{des} proporciona suficiente rango de búsqueda)

LZ77+Rice: la situación no mejora

Pero!

Tamaño raw = 34969 bits

Tamaño Rice = 21978 bits

⇒ **Utilizad Rice!**