

Ejercicio 6 y 7 : CC50Q - Teoría de la Información y Redes Neuronales

Prof.: Pedro Ortega <peortega@dcc.uchile.cl>
Aux.: Francisco Claude <fclaude@dcc.uchile.cl>

9 de septiembre de 2005

- Entrega: Viernes, 30 de Septiembre, hasta las 23:59 horas -

Implementación de un (des-) compresor

El objetivo de esta tarea consiste en el diseño e implementación de un *compresor* y *descompresor* de archivos utilizando códigos de *Huffman*. El *compresor* debe a) leer todo el archivo, b) calcular las frecuencias empíricas para cada byte (símbolo) encontrado, c) construir un código de *Huffman* e imprimir su derivación a la salida estándar y d) codificar el archivo dado dentro de un *archivo binario*. El *descompresor* debe reconstruir al archivo original a partir del archivo codificado.

Los archivos comprimidos deben tener el mismo nombre pero con extensión `.huff`. Ambos programas *deben respetar* el siguiente formato para su invocación en la línea de comandos:

```
> encode <archivo>  
> decode <archivo-comprimido>
```

Puede implementar sus programas en Java o en C. Entregue su tarea en formato `zip` ó `gzip` a través de *U-Cursos*. Ésta debe consistir en: a) código fuente, b) los programas compilados para correr en Windows ó Anakena y c) una brevísima explicación de uso.

La evaluación se realizará (automáticamente) comprimiendo y descomprimiendo archivos de prueba. *Si algún archivo no se recupera en forma intacta, su nota final será 1.*

Notas:

1. Conviene almacenar el código de *Huffman* en la cabecera del archivo para poder reconstruirlo en el descompresor.
2. Recuerde que sólo puede almacenar byte a byte. Dado que los códigos pueden ser de largo variable (fracciones de bytes), deberá inventar una estrategia para empaquetar los bits dentro de bytes para codificar el archivo.
3. Calcule la entropía del alfabeto y estime el largo del archivo resultante. Compárelo luego con el tamaño del archivo comprimido para validar su algoritmo.
4. Haga una implementación sencilla. *Esta tarea le tomará mucho tiempo.*