



Ejercicios de Algoritmia Básica (30229)

Grado en Ingeniería Informática.

Curso 2021-2022

Universidad de Zaragoza
Escuela de Ingeniería y Arquitectura
Departamento de Informática e Ingeniería de Sistemas
Area de Lenguajes y Sistemas Informáticos

21 de febrero de 2022

Organización general de las prácticas.

Equipos de trabajo

Se formarán equipos de dos personas. Si una de las dos abandona la asignatura, la otra deberá terminar en solitario. Se dispone hasta el día **1 de marzo** para indicar a través de Moodle (“formación de equipos de prácticas”) los integrantes de cada equipo. Transcurrido ese plazo, los profesores formarán los equipos restantes con las personas sin asignación conocida y se comunicarán tales equipos.

Entrega de la práctica

Las entregas de prácticas se realizarán en el computador `hendrix`.

- La entrega de la práctica mediante la ejecución de:

```
someter ab_21 practica.tar
```

La fecha **límite es el 15 de mayo**.

El fichero `practica.tar` contendrá **un directorio denominado `practica_NIA1_NIA2`** (siendo NIA1 y NIA2 los números identificadores de cada estudiante asignados por la Universidad de Zaragoza, y NIA1 será el NIA menor. En el caso de un grupo de prácticas formado por un único alumno, la carpeta tendrá como nombre `practica_NIA` con el identificador de ese alumno) con los ficheros de texto incluyendo:

- Descripción general del programa: cómo está organizado, qué se puede y qué no se puede hacer (tiene que llamarse **LEEME**).

Contendrá en sus primeras líneas la lista de integrantes del grupo, con el siguiente formato:

```
Apellido1 Apellido2, Nombre [tab] correo@electronico [tab] login en hendrix
Apellido1 Apellido2, Nombre [tab] correo@electronico [tab] login en hendrix
```

en orden alfabético. Donde [tab] representa el caracter tabulador.

- Listados del código debidamente comentados y dispuestos para ser compilados y utilizados. Deberán seguir una estructura lógica para poder encontrar y navegar adecuadamente cada una de las partes de la práctica.
- Un programa para la *shell* denominado **ejecutar.sh** que automatice la compilación y ejecución de los programas entregados con algunos casos de prueba. Deberá funcionar en *hendrix*. Idealmente, este *script* llamaría a otro por cada una de las partes de la práctica, que deberían poder ejecutarse de manera independiente.
- Los ficheros auxiliares de entrada necesarios para ejecutar las pruebas del punto anterior.
- Un fichero en formato PDF con el análisis de las pruebas realizadas (máximo 3 páginas, sin portada). Indicar: nombre, apellidos y NIA de cada miembro del grupo de prácticas.

Evaluación

- En la calificación se tendrán en cuenta los siguientes aspectos: documentación, funcionamiento, implementación, diseño de tests de pruebas, análisis de las pruebas realizadas y facilidad para la repetición de las pruebas por los profesores.
- Se aplicarán las reglas de tratamiento de casos de plagio explicadas en la presentación de la asignatura.

Práctica

La práctica este año se relaciona con la algoritmia y la gestión de información. En particular, el almacenamiento eficiente de la misma, con dos aspectos diferentes.

- El primero se refiere al almacenamiento de los propios datos, tratando de ahorrar espacio; consistirá en la aplicación del algoritmo voraz de Huffman.
- El segundo aspecto requiere el almacenamiento del historial de un fichero, de manera eficiente; para ello se conserva una versión del fichero y los cambios necesarios para llegar a otras versiones. Se utilizará programación dinámica.

Almacenamiento eficiente de información

¿Qué hay que hacer?

Se debe desarrollar un programa compactador/descompactador huf que implemente el todo de compactación de ficheros basado en el código de Huffman. La forma de ejecución deberá ser la siguiente:

```
hendrix2.$ huf -c <nombre de fichero>
```

para compactar, donde <nombre de fichero> es el nombre de un fichero cualquiera (de texto o binario), y el programa generará el fichero compactado <nombre de fichero>.huf.

```
hendrix2.$ huf -d <nombre de fichero>
```

Para descompactar, donde <nombre de fichero>.huf guarda un fichero compactado con el programa huf.

Almacenamiento eficiente de los cambios en la información

Los sistemas de control de versiones se utilizan en sistemas de gestión del código de programas, wikis, procesadores de texto, ... Habitualmente almacenan la última versión del documento de texto que estamos editando y los cambios necesarios para obtener a partir de él la versión anterior (y de ésta, a su vez, su anterior, y así sucesivamente).

En esta práctica vamos a implementar un sistema de gestión de versiones para documentos de texto basado en programación dinámica, con decisiones de diseño abiertas: el programa deberá ser capaz de ofrecernos la versión que solicitemos del documento (sería interesante que se pueda obtener una lista de versiones disponibles, por ejemplo). Para ello lo almacenará junto con sus cambios de la forma que se estime más conveniente.

Utilizando la programación dinámica

Para resolver el problema vamos a utilizar las técnicas de programación dinámica.

Nos basaremos en lo aprendido en clase para transformar una cadena en otra, recordando que las operaciones básicas disponibles, para un par de cadenas cualesquiera serían:

- **Substitución** de un caracter por otro.
- **Inserción** de un caracter en la cadena.
- **Borrado** de un caracter de la cadena.

Una de las decisiones que habrá que tomar es cómo se decide qué cadenas emparejar, que puede ser tan simple (o complicado) como se quiera.

¿Qué hay que hacer?

El programa, que podemos llamar `version`. Se entregarán los ficheros necesarios para su correcta compilación y ejecución.