



Primera práctica: *un algoritmo de ordenación por distribución (en cierto modo, es otra forma de divide y vencerás)*

## 1. Consideraciones generales

### Equipos de trabajo

- Las prácticas se realizarán en equipos de dos personas. Si una de las personas abandona la asignatura, la otra deberá terminar en solitario.
- Se dispone hasta el **28 de febrero** para indicar a través de *Moodle* 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

- La entrega de la práctica 1 se realizará en *hendrix*, introduciendo todo lo desarrollado en un subdirectorio llamado *practical*, generando un *.tar* de ese subdirectorio llamado *practical.tar*, y entregándolo mediante:  

```
hendrix:> someter ab_19 practical.tar
```
- La fecha límite de entrega es el **31 de mayo** para la 1ª convocatoria y el **1 de septiembre** para la 2ª.
- La descompresión de *practical.tar* mediante `tar -xvf prácticoal.tar` debe generar un directorio denominado *practical* que contenga los siguientes ficheros:
  - Ficheros fuente debidamente comentados.
  - Instrucciones para realizar la compilación, la ejecución y la repetición de todas las pruebas realizadas.
  - Conjunto de pruebas realizadas (ficheros necesarios para repetir dichas pruebas).
  - Un fichero en formato PDF con la descripción de las decisiones adoptadas y análisis de resultados de las pruebas (máximo 3 páginas). Indicar: nombre, apellidos y NIP de cada miembro del grupo de prácticas.

### Evaluación

- Se considerarán los siguientes aspectos: diseño e implementación de los algoritmos, código claro y comentado, corrección, diseño de pruebas, análisis de las pruebas realizadas, facilidad para la repetición de las pruebas por los profesores, y calidad del documento de descripción y análisis de los resultados.
- Se pretende que cada línea de código presentada haya sido escrita por vosotros. Cualquier caso de copia entre dos equipos distintos supondrá el suspenso para ambos. Si copiáis partes del código de Internet o de algún libro, debéis mencionarlo claramente en vuestro código, de lo contrario se suspenderá la práctica. Por supuesto, las partes de código copiadas no aumentarán la nota, mientras que sí podrán hacerlo las partes de código originales.

## 2. Enunciado de la práctica 1

Se parte de una secuencia de identificadores de productos, representados por códigos numéricos positivos. El número de dígitos que forman cada código debe ser un valor parametrizable. Por ejemplo, estableciendo códigos de 3 dígitos, podríamos tener códigos entre el 000 y el 999; con códigos de 4 dígitos, entre el 0000 y el 9999, etc.

- a. Se debe implementar un algoritmo de *ordenación radix*<sup>1</sup> (concretamente el *least significant digit radix sort*) para ordenar una secuencia de identificadores de productos (con un número dado de dígitos) almacenados en un vector.
- b. A continuación, se debe realizar un estudio comparativo de la eficiencia del algoritmo implementado con respecto a la ordenación proporcionada por la función *sort()*<sup>2</sup> de C++. Para ello, habrá que ejecutar ambos algoritmos en distintas situaciones, variando dos parámetros: el número de identificadores de productos y el número de dígitos de los códigos de los identificadores. Para facilitar ese proceso, se desarrollará una función que permita generar vectores aleatorios de códigos de productos, dados el número de identificadores y el rango de código. Se harán las pruebas hasta los mayores tamaños posibles y se presentarán los resultados comparativos obtenidos en *hendrix*.

<sup>1</sup> Véanse, por ejemplo, las diapositivas 152 a 157 de <http://webdiis.unizar.es/asignaturas/TAP/material/2.4.digitales.pdf>.

<sup>2</sup> Véase, por ejemplo, <http://www.cplusplus.com/articles/NhAORXSz/>.