

Ejercicios Prácticos Introductorios

Estos ejercicios sirven de introducción mientras se imparten los conceptos necesarios en las clases magistrales. Lo que aprendas, así como los comentarios y consejos que recibas en las sesiones prácticas, te serán de utilidad en las posteriores prácticas de la asignatura.

Se plantean dos ejercicios diferentes, y se pide:

- **Desarrollar** el código fuente en Java, **empleando la clase BigInteger/BigDecimal y sus métodos** (se puede emplear otro lenguaje de programación¹), necesario para **llevar a cabo** los ejercicios prácticos.
- **Documentar** ambos ejercicios mediante sendas memorias con el código fuente, los ficheros por lotes (*batch* o *script files*), ficheros de datos, y gráficas, etc, necesarios.

EJERCICIO PRÁCTICO 1. Cálculo del máximo común divisor de 2 números.

Se pide programar y evaluar dos formas alternativas de determinar los factores comunes a dos números.

1. **Desarrolla** el código fuente para los algoritmos que se indican a continuación:
 - a. Algoritmo 1: Descomposición de los números en sus factores primos, y multiplicación de los factores comunes con el menor exponente.
 - b. Algoritmo 2: **Método de Euclides**.

Las entradas son dos números naturales. La salida es el máximo común divisor.

2. **Determina experimentalmente** (empíricamente) la complejidad computacional de ambos algoritmos. En ambos casos debes realizar el análisis considerando como **tamaño de entrada** el **número de dígitos** de las dos entradas. Es decir, determina la complejidad realizando pruebas de ejecución (desde valores pequeños a los valores máximos que te permita la representación de números según los tipos empleados), midiendo el tiempo empleado en la ejecución. Puedes elaborar los scripts o ficheros por lotes que necesites para generar los datos y/o los gráficos. Además, puedes emplear el lenguaje de programación o *framework* (e.g. Matlab, R, Python, etc.) que desees para generar los datos y capturar los resultados.

Importante: Para generar la gráficas, puedes emplear el lenguaje de programación o *framework* (e.g. Matlab, R, Python, etc.), o herramientas SW (e.g. excel), que muestren los resultados, es decir, tamaño de entrada (eje de abscisas) frente al tiempo de ejecución (eje de ordenadas).

Detalla razonadamente la experimentación realizada y las conclusiones obtenidas.

¹ Se puede emplear otro lenguaje de programación, pero con las clases, métodos, etc. que permitan las mismas funcionalidades que la clase *Big Integer* de Java: realizar cálculos con un número de dígitos [mucho] mayor que el permitido por los tipos de datos estándar (o primitivos) del lenguaje.

Debes llevar a cabo un número de pruebas suficiente para obtener resultados significativos que te apoyen en las conclusiones. Es conveniente que las gráficas tengan sentido, sean significativas, etc. Plantea alternativas para obtener resultados lo más fiables posibles. En caso de duda, consulta con tu profesor.

3. **Realiza una evaluación analítica de ambos algoritmos**, aplicando la teoría vista en clase. Explica razonadamente tus argumentos. ¿Qué conclusiones obtienes de este estudio analítico? Compara los resultados del estudio experimental y del estudio analítico, ¿qué conclusiones obtienes?
4. Presenta tus resultados en una memoria, en forma académica: (a) Resumen, (b) Introducción, (c) Desarrollo, (d) Conclusiones, (e) Referencias. No olvides identificar tus tablas y gráficas con un número y un título.

EJERCICIO PRÁCTICO 2. Cómputo de la *primalidad* de un número natural.

Desarrolla en código fuente un algoritmo, **que tú mismo diseñes**, que compute si un número natural es primo o compuesto. Puedes diseñar varios algoritmos (o versiones, mejoras, de un mismo algoritmo).

De la misma manera que en el Ejercicio Práctico 1 obtén analíticamente y experimentalmente la complejidad computacional del (los) algoritmo(s) que implementes en tu código fuente. Detalla, razonadamente, la experimentación realizada y las conclusiones obtenidas.

Presenta tus resultados en una memoria, con las características descritas en el ejercicio práctico 1.

ENTREGAS

En Aula Global tendrás entregadores semanales que utilizaremos para que puedas dejar constancia de vuestro trabajo en la evaluación continua. En cada uno encontrarás la fecha límite de entrega. En el cronograma en Aula Global dispondrás de una referencia respecto a las entregas previstas.

Las indicaciones a continuación se refieren a la entrega final. En las entregas parciales puedes incluir el material que tengas disponible hasta el momento.

Los ficheros que debe contener vuestra entrega deben ser los siguientes:

- Código fuente y ficheros por lotes desarrollados. Haz referencia a ellos como anexos de la memoria reporte (incluso puedes incluir en la memoria el código fuente).
- Ficheros de datos (csv, Excel, etc.) que hayas obtenido de las pruebas de ejecución para generar los gráficos.
- Gráficos (en ficheros de imagen png), y ficheros cuya información y descripción sea auto-contenida en el propio gráfico (títulos, leyenda, etiquetas para los ejes, etc.). Si empleas scripts de R o Matlab para generar los gráficos, inclúyelos también.

- Código fuente y ficheros por lotes (batch scripts en Windows, bash script en Linux), o scripts de R o Matlab que hayas podido desarrollar. Haz referencia a ellos en el apartado de Referencias (Bibliografía), o inclúyelos si no son muy extensos como anexos en la memoria.
- Una memoria (formato pdf) que incluya:
 - Las pruebas experimentales que hayas realizado para determinar empíricamente el coste computacional de los algoritmos, y su complejidad computacional.
 - El estudio analítico de la complejidad computacional de los algoritmos.
 - Las complicaciones y/o dificultades que hayas encontrado, y el detalle de cómo las has podido resolver.
 - Conclusiones generales.

El fichero comprimido deberá nombrarse como:

EP<X> - <GP>-Apellido1-Apellido2-Apellido3.zip, siendo *ApellidoK* el primer apellido, en orden alfabético, de cada uno de los integrantes del equipo. <X> representa el número de entrega: 1 para el hito 1, y F para la entrega final. <GP> representa vuestro número de grupo de prácticas, comprobadlo en el listado (no es el grupo reducido). Esta forma de nombrar el fichero será utilizará también en las siguientes entregas.

Notas importantes:

- Las tareas requeridas para las siguientes prácticas serán similares.
- Los grupos de trabajo serán de tres personas.