

**A. Contextualización**

Las técnicas generales de diseño de algoritmos ofrecen un amplio y variado conjunto de herramientas que se pueden usar de forma flexible en la resolución de muchos tipos de problemas. La aplicación de una técnica debe entenderse como un proceso metódico, que empieza con la interpretación y el modelado del problema desde la perspectiva de esa técnica, sigue con la definición de las partes genéricas del esquema algorítmico (tipos de datos y funciones básicas), y acaba en la implementación, el refinamiento y la optimización del algoritmo.

Lógicamente, el paso previo es determinar qué técnica resulta más adecuada para el problema de interés. En algunos casos se podrán aplicar varias técnicas, a veces una técnica se podrá aplicar de formas distintas, y en otros casos ninguna técnica estándar será viable, debiéndose buscar una solución *ad hoc*.

**B. Enunciado del problema**

Cada grupo deberá escoger tres problemas, dentro del conjunto de problemas que le corresponda, y resolverlo usando algunas de las siguientes técnicas:

- Avance rápido.
- Programación dinámica.
- Backtracking con esquema iterativo, como el visto en clase.

Como mínimo, se debe usar una vez cada una de las técnicas. Además, todos los problemas deben estar resueltos de forma óptima. Esto significa, por ejemplo, que si para un problema se diseña un algoritmo voraz que no encuentra la solución óptima, se deberá programar otro algoritmo de backtracking o programación dinámica óptimo.

Los programas se deben **ajustar rigurosamente a los formatos de entrada y salida** especificados en los enunciados. No incluir ningún interface para los programas. En todos los casos, se usará para la entrada/salida del programa la entrada y salida estándar (leer de teclado, escribir en pantalla). Se deberá comprobar la corrección de los programas, mostrando la ejecución sobre diversos ejemplos. La ejecución de estos ejemplos se hará redireccionando la entrada y salida estándar a ficheros:

```
>> ./programa <prueba1.txt >salida1.txt
```

Aplicar el proceso de diseño de las técnicas visto en clase (interpretación del problema, esquema general, funciones básicas del esquema, etc.), documentando las decisiones tomadas en cada punto. Hacer un estudio de los tiempos de ejecución de los algoritmos diseñados. El estudio debe ser breve pero significativo (es decir, usar tamaños suficientemente grandes). Extraer conclusiones relevantes sobre la eficiencia y sobre el propio proceso de aplicación de las técnicas.

**B.1. Número de los problemas a resolver**

Se deberá considerar la suma de los números de DNI de los miembros del grupo. Sea **P** la suma de los DNI de los miembros. El grupo de problemas a resolver está en función del valor de **E** obtenido con la fórmula: **E = P mod 5**.

<b>E = P mod 5</b>	<b>Número de problemas</b>
0	10602, 10405, 704, 10382
1	10131, 529, 10160, 10382
2	357, 10032, 10672, 704
3	10270, 10249, 10100, 10051
4	410, 10422, 10405, 10066

Los alumnos deberán elegir y resolver 3 problemas dentro del grupo que les corresponda. Los enunciados de los problemas se encuentran disponibles en la página web de la asignatura (<http://dis.um.es/profesores/ginesgm/aed.html>).

## **B.2. Validación automática de los programas**

Para validar la corrección de los programas se puede utilizar el juez on-line de la Universidad de Valladolid (<http://acm.uva.es>). Los números de los problemas coinciden con los códigos de problemas en este juez. Se valorará **muy positivamente** que los programas diseñados sean aceptados por el juez on-line, aunque no será un requisito indispensable para aprobar. No obstante, hay que recordar que en esta práctica no basta con escribir los programas: hay que documentar el proceso de análisis y diseño, y realizar el estudio de la eficiencia.

## **C. Memoria de la práctica**

La memoria de la práctica deberá contener obligatoriamente los siguientes apartados.

### **C.1. Portada**

Nombre de los alumnos, número de prácticas y e-mail de cada uno. En la cuenta de prácticas se debe incluir la memoria de prácticas (en formato DOC, PS o PDF).

### **C.2. Resolución de problemas**

Para cada problema resuelto por el grupo se deberán incluir los siguientes apartados.

#### **C.2.1. Análisis y diseño**

Describir el proceso de aplicación de la técnica, el esquema algorítmico utilizado, así como otras cuestiones y decisiones de diseño que se consideren de interés.

#### **C.2.2. Listado del código**

Código fuente del programa o programas implementados, en C/C++.

#### **C.2.3. Ejemplos de uso y eficiencia**

Se deberán mostrar brevemente algunos ejemplos de aplicación. Asimismo, se deberá hacer un pequeño estudio de la eficiencia conseguida.

### **C.3. Conclusiones**

Extraer conclusiones relevantes comparando las diferentes técnicas, en cuanto a su proceso de aplicación y a la eficiencia y resultados conseguidos.

### **C.4. Informe de desarrollo**

Igual que en las prácticas 1 y 2. Recordar las conclusiones y valoraciones personales.

## **D. Evaluación de la práctica**

### **D.1. Obligatorio**

Para aprobar esta práctica se requiere que:

- Los programas que sean desarrollados se puedan compilar sin errores y funcionen correctamente en las máquinas del laboratorio de prácticas.
- Los problemas asignados al grupo de prácticas deben estar resueltos correctamente o de forma óptima (en el caso de los problemas de optimización).
- La memoria de la práctica debe contener todos los puntos indicados en el apartado C. La memoria debe ser entregada en el plazo que se establezca.
- Todos los programas deben ser originales (obviamente). La manipulación o copia supondrá no superar la práctica.

## **D.2. Criterios de valoración**

La práctica se puntuará de acuerdo a los siguientes criterios:

- Aplicación de las técnicas y las ideas vistas en clase.
- Correcto diseño de los algoritmos y uso del lenguaje.
- Comprobación de la corrección de los algoritmos, valorándose muy positivamente la aceptación por el juez on-line.
- Eficiencia de los algoritmos diseñados.
- Significatividad de la comparativa de eficiencia y las conclusiones extraídas.

## **D.3. Otras cuestiones**

La práctica se deberá realizar en grupos de dos alumnos.

Todos los ficheros generados (incluidos los programas y los casos de prueba) y la memoria de prácticas deben estar accesibles en un subdirectorio **practica3** dentro de la cuenta del grupo de prácticas.

Se establece como fecha tope de entrega de esta práctica el lunes 20 de junio de 2005.