

A. Contextualización

Las **técnicas generales de diseño de algoritmos**, estudiadas en la segunda parte de la asignatura, ofrecen un amplio y variado conjunto de herramientas que se pueden usar en la resolución de muchos tipos de problemas. La aplicación de una técnica concreta 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 con la implementación, prueba, refinamiento y 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 dos problemas, dentro del conjunto de problemas que le corresponda, y resolverlos usando las tres siguientes técnicas:

- Avance rápido.
- Backtracking con esquema iterativo, como el visto en clase.
- Ramificación y poda, con una estrategia LC.

Los programas se deben **ajustar rigurosamente a los formatos de entrada y salida** especificados en los enunciados. No incluir ningún interface adicional 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 y modelado del problema, esquema general del método, funciones básicas del esquema, etc.), documentando las decisiones tomadas en cada punto. Hacer un estudio comparativo de los tiempos de ejecución de los algoritmos diseñados para diferentes tamaños. El estudio debe ser breve pero significativo (es decir, usar tamaños suficientemente grandes). Extraer conclusiones relevantes sobre la eficiencia, la comparación entre técnicas y sobre el propio proceso de aplicación de las técnicas. ¿Son comparables los algoritmos resultantes de las distintas 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 los problemas a resolver está en función del valor de **E** obtenido con la fórmula: **E = P mod 5**.

E = P mod 5	Número de problemas
0	10602, 704 (908), 10382, 529 (909)
1	529 (909), 10160, 11045 (907), 10672
2	10672, 10661, 1032 (906), 704 (908)
3	10270 (910), 10249, 10051 (801)
4	410, 10422 (911), 10664, 529 (909)

Los alumnos deberán elegir y resolver 2 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/files/doc/aed/pract3.zip>).

B.2. Validación automática de los programas

En la tabla de arriba, los números que no están entre paréntesis corresponden al identificar del problema en el juez on-line de la Universidad de Valladolid (<http://icpcres.ecs.baylor.edu/onlinejudge/>). Los números entre paréntesis son el mismo problema en el juez on-line de la asignatura (<http://dis2.um.es/~mooshak>, concurso “AED: Backtracking”, y para el 801 “AED: Programación dinámica”).

- Para los segundos (los que están en Mooshak), se requiere que al menos una de las tres versiones implementadas sea aceptada por el juez.
- Para los primeros (los de Valladolid), no es imprescindible la aceptación en el juez on-line de Valladolid, pero se valorará **muy positivamente** que se obtenga dicha aceptación.
- Puesto que esta práctica es en grupos, si los alumnos están realizando las actividades de evaluación continua de los temas 5 y 6, los ejercicios elegidos no podrán coincidir con los de dichas actividades.
- Además, hay que recordar que en esta práctica no basta con escribir los programas: hay que documentar el proceso de análisis y diseño, explicar la aplicación de las técnicas generales, y realizar el estudio comparativo entre técnicas.

B.3. Material adicional

Como se ha comentado, es obligatorio ajustarse a los esquemas de backtracking y de ramificación y poda estudiados en la asignatura. Esto requiere, en el segundo caso, implementar una **lista de nodos vivos**. Se ofrece a los alumnos una implementación del tipo “lista de nodos vivos” (que, en esencia, es simplemente una cola de prioridad) en:

<http://dis.um.es/profesores/ginesgm/files/doc/aed/tema10/Inv.cpp>

Sugerimos que los alumnos utilicen este tipo. Evitar complicaciones en el código: definir el tipo nodo con memoria estática, declarar la variable LNV de forma estática, también las variables o parámetros de tipo nodo, definir funciones para generar, solución, etc. dentro del esquema.

C. Memoria de la práctica

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

C.1. Portada

Nombre de los alumnos y e-mail de cada uno.

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, representación de la solución, funciones básicas del esquema, 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. Indicar si el programa ha sido aceptado o no por el juez on-line. Asimismo, se deberá hacer

un pequeño estudio teórico/práctico de la eficiencia conseguida, comparando las diferentes técnicas sobre el mismo problema.

C.3. Conclusiones

Extraer conclusiones relevantes comparando las 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, valoraciones personales y estimación del tiempo en completar la práctica.

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 y 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 establecido.
- Todos los programas deben ser originales. 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 para los problema del juez on-line de Valladolid.
- 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. Excepcionalmente se puede hacer individualmente. En ese caso, bastará con resolver un solo problema por los tres métodos. En cualquier caso, los problemas resueltos no podrán coincidir con los entregados para las actividades de evaluación continua.

Todos los ficheros generados (incluidos los programas y los casos de prueba) y la memoria de prácticas deben ser entregados en formato electrónico (disquete, CD, email) al profesor correspondiente.

Se establece como fecha tope para la entrega de esta práctica el lunes 23 de junio de 2008.