



GRADO

GUÍA DE ESTUDIO DE LA ASIGNATURA PROGRAMACIÓN Y ESTRUCTURAS DE DATOS AVANZADAS



2011-2012

GRADO EN INGENIERÍA en INFORMÁTICA y en TECNOLOGÍAS DE LA INFORMACIÓN

1.- PLAN DE TRABAJO

Este plan de trabajo tiene como objetivo guiar al alumno en el estudio de la asignatura y ayudarle a conseguir las habilidades y destrezas que debe haber adquirido tras el estudio de la asignatura y que se han descrito en la primera parte de la guía de estudio. Presentamos en primer lugar el programa detallado de la asignatura, para pasar después a presentar un cronograma con la distribución de estos temas a lo largo del semestre.

1.1 Contenido y Programa de la asignatura

La asignatura se estructura en los siguientes bloques temáticos:

- Introducción (tema 1 del texto base)
- Estructuras de datos avanzadas (tema 2 del texto base)
- Esquemas algorítmicos (temas 3,4,5, 6 y 7 del texto base)

El primer tema está dedicado a la introducción, en el que se presenta el contexto de la asignatura, resaltando la necesidad de crear programas fiables. Se describe el concepto de esquema algorítmico y se identifican los conocimientos que debe poseer el alumno para poder seguir adecuadamente la asignatura.

A continuación, se estudia el tema dedicado a tres tipos de estructuras de datos avanzadas, en concreto: los grafos, los montículos y las tablas hash. Tanto los grafos como los montículos se utilizarán como estructuras fundamentales en algunos esquemas algorítmicos que se verán posteriormente. El principal objetivo de este capítulo con respecto a los grafos y los montículos, es que el lector sea capaz de utilizarlos de forma adecuada y eficiente en los esquemas algorítmicos en los que sean necesarios. Para ello, debe entender su estructura y las operaciones fundamentales. Con respecto a las tablas hash, aunque su estudio y aplicación se limita a este capítulo, se pretende que el lector comprenda su utilidad y conozca las principales funciones hash y de resolución de colisiones asociadas a esta estructura, para poderla aplicar a los problemas en los que es necesaria.

El último bloque temático, dedicado a los esquemas algorítmicos, comprende los capítulos dedicados a los esquemas voraz, divide y vencerás, programación dinámica, vuelta atrás y ramificación y poda. El esquema voraz (greedy algorithms) se aplica a problemas de optimización en los que la solución se puede construir paso a paso sin necesidad de reconsiderar decisiones ya tomadas. El esquema de divide y vencerás se basa en la descomposición de un problema en subproblemas de su mismo tipo, lo que permite disminuir la complejidad y en algunos casos, paralelizar su resolución. La técnica de programación dinámica se caracteriza por registrar resultados parciales que se producen durante la resolución de algunos problemas y que se utilizan repetidamente. El esquema de vuelta atrás o retroceso se aplica a problemas en los que sólo podemos recurrir a una búsqueda exhaustiva, recorriendo el espacio de todas las posibles soluciones hasta que encontremos una de ellas o hasta que hayamos explorado todas las opciones, concluyendo así que no existe la solución buscada. Puesto que esta búsqueda exhaustiva es muy costosa, es importante aplicar el conocimiento disponible sobre el problema para terminar la exploración de un camino tan pronto como se sepa que dicho camino no puede alcanzar una solución. El esquema de ramificación y poda que se emplea en aquellos problemas en los que el objetivo es la optimización de uno o más criterios en la solución alcanzada. Este esquema será menos eficiente que el voraz, cuando ambos se pueden utilizar para realizar la optimización, por lo que su aplicación está indicada sólo cuando no se conoce un algoritmo voraz válido

para el problema. Al igual que en el caso de esquema de vuelta atrás, se exploran todas las posibles alternativas que permitan llegar a una solución, pero en este caso, además de cortar la exploración de los caminos que no pueden alcanzar una solución, se evita la de aquellos que sólo pueden alcanzar una solución peor que otras, ya que el objetivo es la optimización.

El programa detallado de estos temas es el siguiente:

TEMA 1 Introducción

- 1.1 Antecedentes
- 1.2 Noción de esquema algorítmico

TEMA 2 Estructuras de datos avanzadas

- 2.1 Grafos
- 2.1.1 Definiciones básicas
- 2.1.2 Tipos de grafos
- 2.1.3 Representación de grafos
- 2.1.4 Recorrido de grafos
- 2.1.5 Árboles de recubrimiento
- 2.1.6 Puntos de articulación
- 2.1.7 Ordenación topológica de un grafo dirigido acíclico
- 2.1.8 Camino más corto desde la raíz a cualquier otro nodo
- 2.1.9 Otros algoritmos sobre grafos
- 2.2 Montículos
- 2.2.1 Implementación y operaciones sobre elementos del montículo
- 2.2.2 Eficiencia en la creación de montículos a partir de un vector
- 2.2.3 Otros tipos de montículos
- 2.3 Tablas de dispersión (hash)
- 2.3.1 Funciones Hash
- 2.3.2 Resolución de colisiones

TEMA 3 Algoritmos voraces

- 3.1 Planteamiento General
- 3.1.1 Algoritmos voraces como procedimientos heurísticos
- 3.2 Algoritmos voraces con grafos
- 3.2.1 Árboles de recubrimiento mínimo: algoritmo de Prim
- 3.2.2 Árboles de recubrimiento mínimo: algoritmo de Kruskal
- 3.2.3 Camino de coste mínimo: algoritmo de Dijkstra
- 3.3 Algoritmos voraces para planificación
- 3.3.1 Minimización del tiempo en el sistema
- 3.3.2 Planificación con plazos
- 3.4 Almacenamiento óptimo en un soporte secuencial
- 3.4.1 Generalización a n soportes secuenciales
- 3.5 Problema de la mochila con objetos fraccionables
- 3.6 Mantenimiento de la conectividad
- 3.7 Problema de mensajería urgente
- 3.8 Problema del Robot desplazándose en un circuito
- 3.9 Asistencia a incidencias

TEMA 4 Divide y Vencerás

- 4.1 Planteamiento y esquema General
- 4.2 El puzzle tromino
- 4.3 Ordenación por fusión
- 4.4 Ordenación rápida (Quicksort)
- 4.5 Cálculo del elemento mayoritario en un vector
- 4.6 Liga de equipos.
- 4.7 Skyline de una ciudad

TEMA 5 Programación Dinámica

- 5.1 Planteamiento General
- 5.2 Los Coeficientes Binomiales
- 5.3 Devolución de cambio
- 5.4 El Viaje por el Río
- 5.5 La Mochila
- 5.6 Multiplicación Asociativa de Matrices
- 5.7 Camino de coste mínimo entre nodos de un grafo dirigido
- 5.8 Distancia de edición

TEMA 6 Vuelta Atrás

- 6.1 Planteamiento General
- 6.2 Coloreado de grafos
- 6.3 Ciclos Hamiltonianos
- 6.4 Subconjuntos de suma dada
- 6.5 Reparto equitativo de activos
- 6.6 El robot en busca del tornillo
- 6.7 Asignación de cursos en una escuela

TEMA 7 Ramificación y Poda

- 7.1 Planteamiento General
- 7.2 Asignación de Tareas: Pastelería
- 7.3 El viajante de comercio
- 7.4 Selección de tareas: Cursos de Formación
- 7.5 Distancia de Edición

La numeración de los temas del programa de la asignatura coincide exactamente con la numeración de los temas del texto base:

TEMAS DE PROGRAMA DE LA ASIGNATURA	TEMAS DEL TEXTO BASE
1	1
2	2
3	3

4	4
5	5
6	6
7	7

1.2 Cronograma de estudio de la asignatura

A continuación presentamos una posible ordenación temporal de las actividades que deben realizarse para estudiar la asignatura. Este plan de trabajo es orientativo pero se aconseja su seguimiento para facilitar el aprendizaje de la asignatura, y para hacer un reparto adecuado de la carga de trabajo. La asignatura tiene tanto componentes teóricos como prácticos que es importante compaginar. Dependiendo de cada caso, unos alumnos necesitarán dedicar más tiempo a uno u otro aspecto, por lo que la distribución de tiempos se ha realizado por temas sin distinguir entre el tiempo dedicado al estudio del contenido teórico y el dedicado a la resolución de problemas. Además del estudio de los temas, los alumnos deben realizar dos prácticas obligatorias a las que hemos asignado una semana de tiempo en el cronograma. Sin embargo, hay que tener en cuenta que cada una de estas prácticas se corresponderá con alguno de los temas de la asignatura, por lo debe entenderse que el tiempo asignado a ese tema se repartirá entre el estudio y la realización de la práctica, ya que entendemos que la realización de la práctica debe contribuir a facilitar la comprensión del tema correspondiente.

La planificación que proponemos está estructurada en 16 semanas y de ellas se hace el siguiente reparto entre los temas de estudio:

- Tema 1: es breve y sólo requiere su lectura, ya que presenta la asignatura y la noción de esquema algorítmico.
- Tema 2: abarca el estudio de las estructuras de datos avanzadas, que en algunos casos son necesarias para el desarrollo de los algoritmos y esquemas expuestos, y abarca tres semanas. La semana 1 se abordaría el estudio de los grafos, La semana 2 el estudio de montículos y por último el estudio de las tablas de dispersión (hash) en la 3ª semana.
- Tema 3: se dedica al estudio de los algoritmos voraces. Se planificaría con tres semanas de estudio. Durante la primera semana se realizaría una introducción al esquema y algún ejercicio y durante las otras dos, se abordarían las diferentes familias de problemas que se resuelven mediante este esquema: grafos, planificación y optimización. Es aconsejable que el estudio de este esquema se realice cuando se tienen recientes los conocimientos sobre grafos.
- El Tema 4 está dedicado al esquema Divide y Vencerás y se abordaría en 2 semanas. En este tipo de esquema, la variedad de subfamilias es menor, pero debe hacerse más hincapié en la inducción realizada en las llamadas recursivas, y por tanto el aspecto teórico del esquema debe estudiarse detalladamente antes de abordar la instanciación. La primera semana se reserva para el estudio

teórico y en la segunda se abordan dos o tres problemas en profundidad que ilustren la instanciación del esquema.

- El Tema 5 estudia la resolución de problemas mediante programación dinámica. Se le dedican dos semanas, de manera que como se ha hecho en temas anteriores, se introduzca el esquema y los fundamentos del mismo en la primera semana y se complete con ejercicios en la otra.
- El Tema 6 está dedicado estudio de los algoritmos de backtraking o de retroceso. Hay que tener presente el concepto de grafo y abordar el esquema entendiendo la noción de exploración en un grafo implícito frente al concepto de grafo explícito de la estructura de datos. De esta manera, es importante que se adquiera conciencia de que por un lado la exhaustividad, y por otro la generación implícita del recorrido son los aspectos fundamentales, además claro está, de la capacidad de retroceso en el espacio de búsqueda.
- El Tema 7 aborda el esquema de ramificación y poda, que se aplica a problemas de optimización. Se debe tener fresca la noción de montículo para el almacenamiento de los nodos pendientes de explorar y comprender porqué esta estructura de datos es útil para considerar en cada momento el nodo más prometedor no explorado. Este esquema puede verse como una extensión del anterior, para aplicarlo a problemas de optimización en lugar de búsqueda, por lo que es importante centrarse los elementos nuevos, como son el montículo y la actualización de las cotas superiores e inferiores del objetivo a optimizar. La temporización será en este caso de dos semanas.

FECHAS	TEMAS	ACTIVIDADES
Semanas 1, 2 y 3	Tema1: Introducción	- Lectura del tema 1
	Tema 2: Estructuras de datos avanzadas	- Estudio del contenido teórico del tema 2
		- Resolución de problemas propuestos
Semanas 4,5 y 6	Tema 3: Algoritmos voraces	- Estudio del contenido teórico
		- Resolución de problemas ya resueltos en el texto base
Semanas 7 y 8	Tema 4: Divide y Vencerás	- Estudio del contenido teórico
		- Resolución de problemas ya
		resueltos en el texto base
Semana 9	Primera práctica obligatoria	Entrega al tutor del centro asociado de la primera práctica obligatoria

Semanas 10 y 11	Tema 5: Programación Dinámica	 Estudio del contenido teórico Resolución de problemas ya resueltos en el texto base
Semanas 12 y 13	Tema 6: Vuelta atrás	 Estudio del contenido teórico Resolución de problemas ya resueltos en el texto base
Semanas 14 y 15	Tema 7: Ramificación y poda	 Estudio del contenido teórico Resolución de problemas ya resueltos en el texto base
Semana 16	Segunda práctica obligatoria	Entrega al tutor del centro asociado de la segunda práctica obligatoria

2.- ORIENTACIONES PARA EL ESTUDIO DE LOS CONTENIDOS

Las orientaciones para el estudio de los temas se encuentran desarrolladas en el texto base y en la Primera Parte de la Guía de Estudio, por lo que recomendamos hacer una lectura detenida de dichos textos dónde encontrarán información detallada acerca de los resultados del aprendizaje primero a nivel general de la asignatura y después a nivel particular de cada tema, así como la contextualización de la asignatura dentro de los estudios de Grado y de cada tema dentro de la asignatura. Además, en el texto base y para cada tema también hay una descripción de los contenidos previos necesarios, de los contenidos más relevantes a través de los objetivos que se deben alcanzar tras el estudio de cada tema que, por supuesto, están orientados a la consecución de los objetivos generales de la asignatura.

El plan que recomendamos para el estudio de cada uno de los temas es el siguiente:

- 1º. En el caso del tema 2, dedicado a estructuras de datos avanzadas, debe irse estudiando cada una de las estructuras tratadas, y a continuación, antes de pasar a la siguiente, resolver los ejercicios propuestos para ella
- 2º En los temas dedicados a los esquemas algorítmicos, estudiar el apartado dedicado al planteamiento general, en el que, además de motivar la utilidad del esquema y de describirlo detalladamente, se ejemplifica aplicándolo paso a paso a un problema concreto.
- 3º A continuación se debe pasar a resolver los problemas que ya están resueltos en el texto base aplicando el esquema correspondiente sin mirar la solución propuesta. Después debe comparase la solución propuesta en el texto base y la desarrollada por el alumno, e intentar entender a que se deben la diferencias, si las hay. Deben intentar resolverse de esta forma tantos problemas como sea posible, especialmente si la solución desarrollada por el alumno no se corresponde con la del texto base.

4º Una vez estudiado el tema, el alumno puede pasar a resolver los problemas propuestos en el texto base. Gran parte de estos ejercicios o variantes muy similares están resueltos en la bibliografía complementaria, según se detalla en las notas bibliográficas de cada capítulo.

5º En este punto, el alumno está en condiciones de empezar a realizar la actividad práctica obligatoria, si esta se corresponde con el tema estudiado.

2.1 Actividades complementarias sugeridas

Puede resultar muy útil la implementación de versiones sencillas de los algoritmos, buscando una forma simple de tratar la entrada/salida. Esto es especialmente necesario si hay algún esquema que le resulte al alumno particularmente difícil de entender.

También es útil trazar los algoritmos implementados en las prácticas, no sólo para la depuración sino también para la facilitar la comprensión del algoritmo particular y del esquema al que pertenece.

3.- ORIENTACIONES PARA LA REALIZACIÓN DEL PLAN DE ACTIVIDADES

Es fundamental que el alumno resuelva los problemas del texto base por sí mismo, antes de pasar a contrastar su propuesta con la solución detallada en el texto. También deben resolverse los problemas propuestos, cuya solución en muchos casos se encuentra en la bibliografía complementaría.

Un aspecto importante de la metodología es aplicar los conocimientos adquiridos con el estudio de los temas y la resolución de problemas a la realización de prácticas en las que se programan los algoritmos en un lenguaje de programación concreto, que es Java.

3.1 Evaluación

La evaluación de la asignatura se llevará a cabo a partir de las siguientes pruebas:

- Realización de dos prácticas obligatorias a lo largo del semestre (hasta 2 puntos).
- Realización de un examen teórico/práctico (hasta 8 puntos).

3.2 Prácticas de la asignatura

El trabajo del curso incluye la realización de dos prácticas obligatorias de programación en lenguaje Java. El objetivo de estas prácticas es ayudar al alumno a la comprensión de los temas tratados, así como hacerle ver su aplicación.

El enunciado de las prácticas estará disponible tanto en el curso virtual de la asignatura.

Las prácticas son corregidas por los Tutores de los Centros Asociados. La nota asignada por el tutor podrá incrementar hasta un máximo de 2 puntos (1 por cada práctica) en la nota final de la asignatura. Las notas

de las prácticas no se guardan de un curso para otro, pero son válidas para las convocatorias de febrero y de septiembre. El aporte de las calificaciones de las prácticas a la nota final es el siguiente:

Sobresaliente -> 1 Notable -> 0.75 Aprobado -> 0.5

Las prácticas, por tanto, se organizan en los centros asociados bajo la responsabilidad de cada tutor, por lo que los alumnos deben ponerse en contacto con ellos lo antes posible al comienzo el curso para conocer:

- El calendario de entrega de prácticas y de sesiones presenciales de las prácticas.
- La forma de entrega (correo electrónico, disquete, CD, curso virtual, etc.).

El programa editor recomendado es BlueJ, conjuntamente con el compilador incluido en el JDK. Esto quiere decir que los tutores encargados de las clases de prácticas sólo darán soporte para un entorno de esas características. Una versión para instalar y un pequeño manual de instalación se encuentran disponibles tanto en la página web de la asignatura como en el entorno virtual. Por tanto el uso de otros entornos no garantiza al alumno ningún tipo de soporte en la instalación, configuración o ejecución tanto del entorno como de los programas desarrollados.

Por otro lado, el uso de una versión u otra del JDK puede afectar a la compilación de los programas, ya que de una versión a otra se incluyen novedades, por lo que es importante especificar la versión del JDK que se usó para la realización de la práctica, o usar la especificada por el equipo docente en caso de que la haya.

Por último, los programas deben compilar independientemente del entorno de edición que haya sido usado, debiendo el alumno especificar claramente los pasos a seguir para el correcto funcionamiento de las aplicaciones generadas en las prácticas. De no cumplirse este requisito y en caso de no poder ejecutar una práctica, ésta se considerará suspensa. Cualquier copia en las prácticas dará lugar a un suspenso para todo el curso académico.

3.1.1. Material que debe presentar

El material a entregar para cada una de las prácticas es el siguiente:

- Código fuente adecuadamente documentado y ejecutable.
- Documento pdf con la siguiente información:
 - Datos de la asignatura y del alumno:
 - Nombre y código de la asignatura
 - Título de la práctica
 - Nombre y Apellidos
 - NIF
 - Centro Asociado
 - o Descripción de la estructura de datos o esquema que se utiliza en la práctica
 - o Descripción de la entrada/salida
 - Descripción de los datos de prueba utilizados
 - Otras informaciones que se soliciten en el enunciado

3.1.2. Evaluación de las prácticas

Cada práctica se puntuará sobre 10, siendo el aprobado el 5.

Para que se evalúe la práctica es imprescindible que el programa completo compile y funcione.

En la evaluación que realiza el tutor se tendrán en cuenta los siguientes aspectos:

- Correcta utilización de la estructura de datos o el esquema a utilizar.
- Eficiencia del algoritmo.
- Documentación presentada.
- Calidad del código y estilo de programación.
- Posibles mejoras introducidas por el alumno a los requisitos básicos de la práctica.

3.3 Examen

El examen costará de una serie de problemas y cuestiones teórico/prácticas, que podrían ser de tipo test, sobre los temas de la asignatura.

Para que el examen de un alumno sea calificado deberá haber asistido, como mínimo, a dos sesiones presenciales de prácticas en su centro asociado y haber entregado y aprobado las prácticas obligatorias.

4.- GLOSARIO

Algoritmo. Secuencia finita y ordenada de instrucciones que conducen a la resolución de un problema.

Algoritmos voraces. Esquema algorítmico en el que se seleccionan los elementos más prometedores del conjunto de candidatos hasta encontrar una solución.

Árbol. Estructura de datos no lineal compuesta por un conjunto de nodos, cada uno de los cuales puede estar enlazado a cero o más nodos hijos que descienden del nodo padre.

Clase de complejidad P . Conjunto de los problemas de decisión que pueden ser resueltos en una máquina determinista en tiempo polinómico.

Clase de complejidad NP. Conjunto de los problemas de decisión que pueden ser resueltos por una máguina no determinista en tiempo polinómico.

Complejidad temporal de un problema. Número de pasos, especificados en función del tamaño de la entrada, que se requieren para resolver la instancia de un problema definida por dicha entrada, utilizando el algoritmo más eficiente conocido.

Complejidad de un algoritmo. Número de pasos, especificados en función del tamaño de la entrada, que se requieren para resolver la instancia de un problema definida por dicha entrada, utilizando el algoritmo.

Cota inferior asintótica. Función que sirve de cota inferior de otra función cuando el argumento tiende a infinito. Usualmente se utiliza la notación $\Omega(g(x))$ para referirse a las funciones acotadas inferiormente por la función.

Cota superior asintótica. Función que sirve de cota superior de otra función cuando el argumento tiende a infinito. Usualmente se utiliza la notación O(g(x)) para referirse a las funciones acotadas superiormente por la función.

Divide y vencerás. Esquema algorítmico en el que el problema se divide en subproblemas de la misma naturaleza, cuyas soluciones pueden combinarse para resolver el problema original.

Esquema algorítmico. Técnica de diseño de algoritmos para resolver familias de problemas mediante la instanciación de los elementos del esquema al problema.

Estructura de datos. Organización de un conjunto de datos elementales para facilitar su manipulación.

Grafo. Estructura de datos no lineal compuesta por un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos del conjunto.

Montículo (maximal). Estructura de datos del tipo árbol en la que cada nodo padre tiene un valor mayor que el de todos sus nodos hijos.

Programación dinámica. Esquema algorítmico que se caracteriza por registrar resultados parciales que se producen durante la resolución de algunos problemas y que se utilizan repetidamente, consiguiendo de esta forma reducir el coste de cómputo, al evitar la repetición de ciertos cálculos.

Ramificación y poda. Esquema algorítmico que se aplica a problemas de optimización y que se basa en la exploración de un árbol implícito de soluciones de forma que se van podando tan pronto como es posible aquellas ramas que no pueden llevar a una solución óptima.

Recorrido de un árbol. Sucesión de nodos del árbol, de forma que entre cada dos nodos consecutivos de la sucesión haya una relación de parentesco.

Recorrido en anchura. Recorrido de un árbol en el que se accede a todos los nodos de un nivel antes de pasar a un nodo del siguiente nivel.

Recorrido en profundidad. Recorrido de un árbol en que se accede siempre a un nodo hijo que no se ha recorrido previamente, hasta llegar a una hoja. Entonces se accede al nodo anterior, accediendo al siguiente hijo, y así sucesivamente.

Recorrido en preorden. Recorrido de un árbol en el que se accede primero al nodo en curso y luego cada uno de los árboles hijos.

Recorrido en postorden. Recorrido de un árbol en el que se accede primero a todos los subárboles hijos, que se acceden con el mismo criterio, y después al nodo padre.

Tabla de dispersión (hash). Estructura de datos que asocia claves con valores, permitiendo el acceso a los elementos almacenados mediante una transformación de la clave asociada con una función hash que le asocia una posición en el almacenamiento.

Vuelta atrás. Esquema algorítmico en el que se explora exhaustivamente hasta alcanzar una solución un árbol implícito de posibles soluciones, retrocediendo a un nodo anterior cuando se llega a una rama que no permite alcanzar una solución.