FACULTAD DE INGENIERÍA
DEPARTAMENTO DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIONES

Código-Materia: 09687 - ALGORITMOS Y ESTRUCTURAS DE DATOS.
Requisitos: 09639 – Algoritmos y programación II
Programa – Semestre: Ingeniería de Sistemas - Ingeniería Telemática
Período académico: 2014-1
Intensidad semanal: 5 horas
Créditos: 3

Descripción

En este curso los estudiantes se verán enfrentados a la solución de problemas utilizando estructuras de datos, análisis de algoritmos y algunas técnicas de diseño de algoritmos y . El curso está orientado a generar en el estudiante la habilidad de diseñar e implementar estructuras de datos en memoria principal y que son necesarias para resolver un problema, teniendo en cuenta un conjunto de restricciones y criterios de calidad. Estas estructuras deben integrarse a proyectos que deben desarrollarse siguiendo un proceso sistemático y disciplinado, teniendo en cuenta el análisis, planeación, diseño de la aplicación, utilización de pruebas automáticas y generación de documentación.

Objetivos

General

Solucionar problemas desarrollando aplicaciones que hagan uso de estructuras de datos y algoritmos que permiten resolver problemas clásicos de la computación; partiendo del diseño e implementación de dichas estructuras, aplicando nociones de análisis de algoritmos y siguiendo un proceso sistemático y disciplinado de construcción de software.

Terminales

Al final del curso, se espera que el estudiante esté en capacidad de:

1. Analizar, diseñar e implementar soluciones a problemas de computación empleando algoritmos clásicos y algunas técnicas de diseño de algoritmos apoyados en el análisis de complejidad temporal; siguiendo un proceso sistemático y disciplinado de construcción de software.
2. Diferenciar los tipos de estructuras de datos que se pueden utilizar para modelar los elementos del mundo del problema e implementar los algoritmos que manipulan las principales estructuras de datos lineales, de acceso directo, recursivas y Grafos.
3. Proponer y justificar el diseño de estructuras de datos para resolver un problema, utilizando como argumentos: la complejidad de los algoritmos que implementan las operaciones críticas y el espacio ocupado en memoria.

Objetivos por unidad:

Unidad 1:

Al finalizar esta unidad, el estudiante estará en capacidad de:

- Calcular la complejidad de algoritmos iterativos simples y de algoritmos recursivos simples utilizando la notación O.
- Aplicar las distintas estrategias de ordenamiento y diferenciarlas teniendo en cuenta su complejidad.
- Diferenciar las técnicas de diseño de algoritmos utilizadas en distintos algoritmos de ordenamiento.
- Identificar los casos especiales en los que se puede aplicar algoritmos de ordenamiento lineales .
- Explicar la diferencia entre un algoritmo recursivo y uno iterativo y aplicar recursividad para resolver problemas
- Decidir, con criterios de complejidad, sobre la implementación más eficiente en el uso de los diferentes métodos de búsqueda y ordenamiento.

Unidad 2: Diseño básico de estructuras de datos lineales y de acceso directo

Al final de este nivel el alumno será capaz de:

- Proponer y justificar un diseño para implementar una estructura de datos, siguiendo una metodología que tenga en cuenta un conjunto de restricciones impuestas (tiempo, espacio y flexibilidad).
- Construir y utilizar una clase genérica, en la cual sea posible parametrizar el tipo de los elementos contenidos.
- Utilizar estructuras lineales (Lista) de datos como parte del diseño de la solución de un problema.
- Explicar el concepto de iterador y diseñar las estructuras de datos adecuadas para su implementación.
- Diseñar, adaptar y utilizar estructuras de datos de acceso directo por llave, las cuales están basadas en la capacidad de las funciones de hashing de localizar una posición física a partir de una llave lógica.
- Utilizar estructuras tipo colas, pilas y tablas hash como parte de la solución de un problema

Unidad 3: Estructuras recursivas

Al final de este nivel el estudiante será capaz de:

- Implementar y utilizar los métodos static de Java, como una forma de programar las funcionalidades que son independientes del estado de un objeto particular y utilizando recursividad.
- Escribir algoritmos no triviales para el manejo de estructuras recursivas
- Proponer distintas estructuras de datos para representar estructuras recursivas, entendiendo las ventajas e inconvenientes de cada una de ellas.
- Comprender el concepto de orden y balanceo de un árbol, y las consecuencias que esto tiene sobre la complejidad de los algoritmos que manejan esta estructura.
- Utilizar distintos tipos de árboles binarios, n-arios y listas recursivas como parte de la solución de un problema.

Unidad 4: Grafos

Al final de este nivel el estudiante será capaz de:

- Modelar la información de un problema utilizando como estructura de información un grafo dirigido o no dirigido.
- Escribir los algoritmos de búsqueda en grafos necesarios para resolver un problema, utilizando recorridos en profundidad, por niveles y heurísticos. En particular, será capaz de implementar en un lenguaje de programación los algoritmos de búsqueda de caminos más cortos desde una y múltiples fuentes, y de búsqueda de árboles de recubrimiento.
- Diseñar las estructuras de datos más adecuadas para representar un grafo dirigido o no dirigido en memoria principal.
- Utilizar Grafos como parte de la solución de un problema.

Metodología

Para los estudiantes:

La herramienta de E-learning (moodle) es el medio que contiene la información oficial del curso y es responsabilidad del estudiante consultar en ella todo lo referente al curso, especialmente las actualizaciones del material y actividades.

De acuerdo a la metodología de aprendizaje activo de la universidad Icesi, los estudiantes deben preparar, antes de la clase, los temas asignados en la programación del curso. Esto es:

- Leer y analizar el material asignado para la sección de clase. Si no se ha asignado material, entonces investigar sobre los temas acordados en la planeación en la bibliografía recomendada.
- Utilizar estrategias de estudio (mapas conceptuales, mapas mentales, resúmenes, etc) que sean efectivas y que sirvan como refuerzo después del proceso de lectura.
- Contestar las preguntas que contiene el material, así como las preguntas adicionales que el profesor entregue.
- Resolver los ejercicios propuestos en el material, así como los ejercicios adicionales que se le entreguen.
- Formular preguntas que requieran ser resueltas durante la clase.

Durante la clase, el estudiante deberá:

- Plantear las dudas que le quedaron durante el proceso de estudio del tema a tratar.
- Participar en las actividades de revisión y consolidación de conceptos que proponga el profesor.
- Trabajar en la solución de los problemas de aplicación que se propongan.

Después de la clase:

- Establecer las relaciones entre los temas tratados en la clase y el conocimiento previamente adquirido.

- Resolver los ejercicios de aplicación del tema, que tienen un nivel de complejidad mayor al de los ejercicios que resolvió previamente.

Para el desarrollo del curso:

Este curso cuenta semanalmente con dos espacios que son utilizados de la siguiente forma:

- Salón de clases: los estudiantes y el profesor se encontrarán en este espacio, en dos sesiones de una hora y media (una hora, 30 minutos) en las cuales se llevará a cabo la discusión de los diferentes temas y la realización de ejercicios que permitan ponerlos en práctica.
- Sala de cómputo: los estudiantes, el profesor y el monitor comparten el espacio en la sala de computo para llevar a cabo la solución de problemas propuestos y su implementación en el lenguaje Java. Para este componente se han destinado dos (2) horas por semana.

Este curso utilizará Java como lenguaje de programación y Eclipse como entorno de desarrollo. No obstante los objetivos de aprendizaje son independientes del lenguaje y el entorno de programación seleccionado.

Evaluación

Las evaluaciones del curso se han dividido en dos tipos: nota individual y nota grupal. A continuación se explican las actividades que hacen parte de cada uno de los tipos, más adelante encuentra la forma en que será calculada la nota definitiva.

Nota individual:

El curso cuenta con 3 momentos de evaluación individual en el salón de clase:

- **Controles de aprendizaje activo**: corresponde a todas las comprobaciones de lectura y de aprendizaje que se hagan durante el curso. Estas comprobaciones puede ser o no calificables y sin previo aviso por parte del profesor.
- ***Evaluación escrita:*** durante el semestre se llevarán a cabo 3 evaluaciones escritas denominadas exámenes parciales, y examen final. El examen final es acumulativo, contando con un 60% de temas correspondientes a la unidad 4, y un 40% asociada a temas del resto de unidades.
- ***Quiz en sala:*** durante el semestre se llevarán a cabo 3 quices en sala durante las horas del laboratorio.

Supletorios: Los supletorios se realizarán de acuerdo al libro de normas y deberes de la Universidad Icesi. Por tanto las solicitudes, recibos y demás trámites deben realizarse con el profesor del curso en las fechas estipuladas en el reglamento.

Se recomienda consultar en el cronograma del curso las fechas de las evaluaciones escritas y quices en sala.

Nota Grupal:

- ***Miniproyectos***: Al inicio de cada unidad se publicará un ejercicio que abarca todos sus temas. Este trabajo debe ser realizado máximo en grupos de tres integrantes, y deben incluir las plantillas de la metodología PSP. La entrega de cada Miniproyecto será antes del parcial correspondiente a la unidad.
- ***Tareas y talleres:*** Durante el semestre el profesor realizará actividades y asignaciones (tareas) de las cuales algunas serán evaluables según el criterio del profesor.

Se recomienda referirse al documento Reglas de juego cursos Algoritmos para mayor detalle sobre la reglamentación vigente en el curso.

Nota definitiva:

Para calcular la nota definitiva debe considerarse que la nota grupal sólo aplicará para aquellos estudiantes que al final del semestre tengan su nota individual en un valor mayor o igual a 3.0. Teniendo en cuenta lo anterior la nota definitiva será calculada de la siguiente manera:

```
Si (nota individual < 3.0) entonces:
        Nota definitiva = nota individual
Si (nota individual >= 3.0) entonces:
        Nota definitiva = (nota individual * 0.7) + (nota grupal * 0.3)
```

A continuación se especifican los porcentajes de las evaluaciones:

Nota Individual

Evaluación	Porcentaje
Primer parcial	15%
Segundo Parcial	15%
Examen final	25%
Quiz en sala 1	10%
Quiz en sala 2	10%
Quiz en sala 3	10%
Controles de aprendizaje activo	15%
Total nota individual	100%

Nota grupal

Evaluación	Porcentaje
MiniProyecto 1	30%
MiniProyecto 2	30%
MiniProyecto 3	30%
Tareas y talleres	10%
Total nota grupal	100%

Bibliografía

- Sally A. Goldman and Kenneth J. Goldman. *A Practical Guide to Data Structures and Algorithms Using Java.* Boca Raton : Chapman & Hall/CRC, 2009. http://goldman.cse.wustl.edu/
- Mark A. Weiss. *Data Structures and Problem Solving Using Java, 4/E* . Addison-Wesley. 2010
- Steven Skienna. *The Algorithm Design Manual Springer*. 1997
- Thomas Cormen, et al. *Introduction to Algorithms*. The MIT Press 2000
- Jorge Villalobos. Diseño y manejo de Estructuras de Datos en C. McGraw-Hill/ Interamericana Editores. 2006
- Texto de referencia (Algoritmos 1): Jorge Villalobos y Ruby Casallas. Fundamentos de Programación: Aprendizaje Activo basado en Casos. Editorial Prentice-Hall, 2006
- Texto de referencia (Algoritmos 2): Jorge Villalobos. *Introducción a las estructuras de datos: Aprendizaje activo basado en casos*. Editorial Prentice-Hall, 2008.