

UNIVERSIDAD DE PUERTO RICO EN HUMACAO
DEPARTAMENTO DE MATEMÁTICAS

A. Encabezado	Universidad de Puerto Rico en Humacao
B. Nombre del curso	Estructuras de Datos y Algoritmos
C. Codificación	COMP4097
D. Cantidad de horas/créditos	Tres (3) horas contacto ¹ / Tres (4) créditos
E. Requisitos o correquisitos y otros requerimientos	MATE3050 y COMP3082
F. Descripción del curso	

El curso inicia al estudiante a diversas estructuras y algoritmos para la representación y procesamiento de datos. Las estructuras estudiadas incluyen colas, matrices, árboles y grafos. También se estudian algoritmos para el manejo de la memoria de una computadora y algoritmos para la ordenación y búsqueda de datos (internos y externos a la memoria principal). Se hacen análisis elementales de algoritmos, incluyendo el análisis del peor caso y algunos análisis del caso promedio. Estos análisis se hacen con respecto al uso del espacio y con respecto al uso del tiempo. Los estudiantes tendrán la experiencia de realizar un proyecto de complejidad moderada (aproximadamente 2000 instrucciones) y de hacerlo como parte de un equipo de programadores.

G. Objetivos de aprendizaje

Al finalizar el curso, el estudiante estará capacitado para:

1. implementar colas mediante el uso de arreglos y listas enlazadas
2. determinar una manera apropiada de implantar colas para una aplicación dada
3. representar una matriz rala usando cualquiera de varias estructuras posibles
4. determinar ventajas y desventajas de cualquiera de varias estructuras con las cuales es posible representar matrices ralas
5. hacer un análisis de los costos en tiempo y en espacio de varias estructuras utilizables para representar matrices ralas
6. reconocer y representar tipos de información que puede ser representada de forma eficiente con árboles binarios o no binarios
7. implementar recorridos de árboles binarios en preorden, postorden e inorden
8. implementar recorridos de árboles a lo ancho
9. realizar los diversos tipos de recorridos de árboles con uso de recursión, de estibas o de colas
10. implementar árboles de búsqueda binaria con árboles binarios con todas las operaciones primitivas ordinarias (recorrido, inserción, búsqueda, eliminación, determinación de padre)
11. implementar árboles de búsqueda binaria con árboles AVL con todas las operaciones primitivas ordinarias (recorrido, inserción, búsqueda, eliminación)
12. implementar árboles 2-3 u otra clase de árboles balanceados con todas las operaciones primitivas ordinarias (recorrido, inserción, búsqueda, eliminación)

¹ Una hora contacto equivale a cincuenta (50) minutos.

13. usar árboles binarios para resolver algunos problemas como el ordenamiento de una lista de valores, la conversión de una expresión algebraica en un tipo de notación (infija, posfija) a otro tipo de notación, o la obtención de códigos de Huffman
14. definir un digrafo y representarlo con una estructura eficiente
15. definir un grafo no-dirigido y representarlo con una estructura eficiente
16. mencionar varias aplicaciones de grafos y digrafos
17. implementar grafos y digrafos con varias estructuras
18. reconocer ventajas y desventajas relativas en varias de las representaciones de grafos y digrafos
19. implementar al menos dos algoritmos para recorrer un grafo y al menos dos algoritmos para recorrer un digrafo
20. escribir un programa que detecta la presencia de ciclos en un grafo y otro programa que detecta ciclos en un digrafo
21. hallar los componentes fuertes de un digrafo
22. implementar un algoritmo para el ordenamiento topológico de un digrafo
23. escribir programas para obtener un árbol generador de costo mínimo para un grafo cualquiera
24. producir un algoritmo para la obtención de una ruta de costo mínimo entre dos vertientes cualesquiera en un grafo o en un digrafo
25. hacer un análisis en términos de su eficiencia en tiempo de varios de los algoritmos anteriormente mencionados que se aplican a árboles
26. implementar más de una política de manejo de *memoria* (*mejor ajuste, primer ajuste, peor ajuste*)
27. describir aspectos básicos de la tarea de administrar la memoria en una computadora (asignación, reclamación, recogido de basura, compactación, políticas de asignación)
28. realizar una simulación realista del proceso de manejo de memoria que hace una computadora
29. mencionar los aspectos fundamentales que distinguen al ordenamiento externo del interno
implantar un algoritmo de ordenamiento externo

H. Bosquejo de contenido y distribución del tiempo

- Unidad 0: Repaso de metodología de programación - 4 horas
Estilo, criterios de calidad, depuración, tipos abstractos, introducción al análisis de algoritmos
- Unidad 1: Colas, matrices ralas y manejo de memoria - 14 horas
Colas (implantación de colas en vectores circularizados, simulación de procesos de encolamiento, variantes de cola como la de doble encolamiento y la de prioridad), **Matrices ralas** (definición, varias formas de implantarlas, análisis de eficiencia en tiempo y en espacio para varias implantaciones), **Manejo de memoria** (definición de la tarea, bloques de tamaño fijo vs. bloques de tamaño variable, recogido de basura, compactación, políticas de asignación, sistema de panitas)
- Unidad 2: Árboles - 18 horas
Definiciones fundamentales, árboles y árboles binarios, recorridos, implementaciones en vectores y mediante apuntadores, árboles AVL, árboles balanceados, aplicaciones (ordenamientos, conversión entre tipos de notaciones algebraicas, códigos de Huffman, otras)
- Unidad 3: Grafos y digrafos - 18 horas
Definiciones fundamentales, estructuras para representarlos, recorridos, rutas de costo mínimo, árboles generadores de costo mínimo, ordenamiento topológico, detección de ciclos,

componentes fuertes, aplicaciones (problema del agente viajero, redes representadoras de proyectos, otras)

Unidad 4: Ordenamiento externo - 3 horas
Diferencias esenciales con el ordenamiento interno, algoritmos y su análisis de eficiencia en tiempo

La suma de las horas sugeridas es de 57. Las **3 horas** restantes deben ser dedicadas a la administración de evaluaciones parciales.

I. Estrategias Instruccionales

Para lograr los objetivos del curso se realizarán las siguientes actividades instruccionales: conferencias, demostraciones, laboratorios, trabajos en grupo, estudios independientes, discusión de asignaciones. El trabajo de laboratorio es parte integral del curso.

J. Recursos mínimos disponibles o requeridos

La Universidad debe proveer un laboratorio para trabajo independiente de los estudiantes, el equipo electrónico que necesita el profesor para impartir la clase, el programado apropiado para el curso (Intérpretes y Compiladores de todos los lenguajes que se utilizarán, herramientas para el desarrollo de aplicaciones) y acceso a INTERNET.

K. Técnicas de evaluación

3 exámenes parciales, 2 proyectos de programación en equipo, 3 asignaciones y 1 examen final. Cada examen y cada proyecto vale 1/8 parte de la nota, el conjunto de las asignaciones vale 1/8 parte de la nota y el examen final vale 1/4 parte de la nota del curso.

L. Acomodo razonable

Los estudiantes que requieran acomodo razonable deben visitar la Oficina de Servicios para la Población con Impedimentos (SERPI) y comunicarse con el profesor al inicio del semestre para planificar el acomodo necesario conforme a las recomendaciones de SERPI.

M. Integridad académica

El Artículo 6.2 del Reglamento General de Estudiantes de la UPR (Certificación Número. 13, 2009-2010 de la Junta de Síndicos) establece que *“la deshonestidad académica incluye, pero no se limita a: acciones fraudulentas, la obtención de notas o grados académicos valiéndose de falsas o fraudulentas simulaciones, copiar total o parcialmente la labor académica de otra persona, plagiar total o parcialmente el trabajo de otra persona, copiar total o parcialmente las respuestas de otra persona a las preguntas de un examen, haciendo o consiguiendo que otro tome en su nombre cualquier prueba o examen oral o escrito, así como la ayuda o facilitación para que otra persona incurra en la referida conducta”*.

Cualquiera de estas acciones estará sujeta a sanciones disciplinarias en conformidad con el procedimiento disciplinario establecido en dicho reglamento.

N Sistema de calificación

La nota se adjudicará a base de la siguiente escala (porcentual):

100 - 85 A; 89 - 75 B; 74 - 60 C; 59 - 50 D; 5 - 0 F

O. Bibliografía

1. Prichard, J., Carrano, F. M. (2010), *Data Structures and Problem Solving with Java, Walls and Mirrors*. Addison Wesley (3ra Ed.)
2. Kruse, R. , Tondo, C. L., Leung, B., (1996) *Data Structures and Program Design in C*, Prentice Hall (2da Ed.)
3. Kruse, Robert L.; (1994) *Data Structures and Program Design*, Prentice Hall, (2da Ed.)
4. Langsam, Y., Augenstein, M. J., Tenenbaum, A. M., *Data Structures Using C and C++*, Prentice Hall (2da Ed.)

Responsables de las revisiones

- Rev/Elio Ramos/Comité de Currículo, sep 2016
- Rev E Ramos/Comité de Currículo, Mayo 2012
- MacHD/corresp97-98:prontuarios:mate4097/12nov97