



Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area I: Datos

(Programa del año 2017)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
ESTRUCTURA DE DATOS Y ALGORITMOS	ING. EN COMPUT.	28/12	2017	2° cuatrimestre
		026/1		
ESTRUCTURA DE DATOS Y ALGORITMOS	ING. INFORM.	2-	2017	2° cuatrimestre
		08/15		

II - Equipo Docente

Docente	Función	Cargo	Dedicación
LUDUEÑA, VERONICA DEL ROSARIO	Prof. Responsable	P.Adj Exc	40 Hs
REYES, NORA SUSANA	Prof. Co-Responsable	P.Asoc Exc	40 Hs
KASIAN, FERNANDO ANDRES	Responsable de Práctico	JTP Exc	40 Hs
AZAR, ELIANA PAOLA	Auxiliar de Práctico	A.1ra Exc	40 Hs
CASANOVA, CARLOS ANDRES	Auxiliar de Práctico	A.1ra Exc	40 Hs
SOSA TORANZO, CECILIA LORENA	Auxiliar de Práctico	A.1ra Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	2 Hs	1 Hs	3 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
07/08/2017	17/11/2017	15	90

IV - Fundamentación

Se toma en este curso el diseño de estructuras de datos considerando principalmente las evocaciones asociativas con respuesta no múltiple y algunos tipos de evocaciones no asociativas, como las extremales.

Además se inicia al alumno en el análisis de algoritmos, con el fin de que aprenda a comparar distintos algoritmos y porque es fundamental para poder diseñar buenos algoritmos.

En esta asignatura se sientan las primeras bases para construir una base sólida en las disciplinas Algoritmos, Estructuras de Datos y Bases de Datos, de forma tal que se tenga la idoneidad suficiente en la temática contando con los conceptos, principios y teorías que constituyen el ámbito de competencia.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se pretende que el alumno sea capaz de:

- Manejar con idoneidad los conceptos que involucran el diseño de estructuras de datos y algoritmos.
- Conocer algunos de los principales algoritmos y estructuras de datos, incluyendo el análisis de su desempeño.
- Analizar y diseñar algoritmos.
- Desarrollar una actitud crítica frente al uso de las estructuras de datos y algoritmos con los que se pueda enfrentar.
- Desarrollar la actitud de búsqueda de nuevas soluciones.
- Reconocer, modificar, innovar y aplicar los distintos tipos de Algoritmos.
- Frente a una aplicación o problema particular, poder brindar una solución eficiente utilizando los conceptos vistos sobre diseño de estructuras de datos y algoritmos, introducir herramientas de diseño de algoritmos y la ingeniería algorítmica como selección de las estructuras de datos y de las técnicas algorítmicas más adecuadas y además utilizar el análisis de los algoritmos para evaluar y justificar la eficiencia de la solución elegida.

VI - Contenidos

1. Explicitación del objetivo de la materia:

Repaso de temas y homogeneización de terminología. Las estructuras. Dato e información. Doble acepción de estructuras de datos. Visión relacional. Fundamento de la evocación asociativa. Concepto de Servicio en una estructura de datos. Operaciones. Datos de un problema de estructuras de datos. Algoritmos.

2. Evaluación de algoritmos:

Planteo. Problema. Función de costo. Necesidad de una función de evaluación. Propiedades. Medidas en tiempo y espacio. Elección de la función de evaluación. Familias de problemas. Solución parametrizada. Notaciones asintóticas. Propiedades. Complejidad. Clases de Complejidad.

3. Listas, Pilas y Colas:

Listas vinculadas y secuenciales. Incidencia del orden, Incidencia del tipo de recurrencia que la define. Búsqueda binaria. Lista Invertida. Lista de 2 niveles. Pilas y colas. Skip Lists. Operaciones sobre cada tipo de estructura. Aplicaciones.

4. Propiedades, demostraciones y representación de grafos:

Definiciones. Propiedades. Representaciones básicas de grafos. Algoritmos sobre grafos

5. Árboles computacionales:

Definición. Representación computacional. Árbol completo, lleno y balanceado. Árboles Binarios de Búsqueda. Árboles Balanceados en altura. Parva. Operaciones básicas sobre cada tipo de árbol. Aplicaciones.

6. Distribución pseudo-aleatoria de datos:

Motivación. Funciones de pseudo-aleatorización. El problema del rebalse. Distintas propuestas para su manejo. Direcccionamiento Directo, condiciones para su aplicación, relajación de la exigencia de totalidad. Aplicaciones.

7. Técnicas de diseño de algoritmos:

Técnicas ávidas, programación dinámica, dividir para vencer. Ejemplos de aplicación sobre métodos de Ordenamiento. Ejemplos de aplicación en algoritmos sobre Grafos. Otros ejemplos de aplicación de las distintas técnicas.

VII - Plan de Trabajos Prácticos

Las clases prácticas consistirán principalmente de la resolución de prácticos de laboratorio que implementan los conceptos teóricos desarrollados en la asignatura, de manera que pueda poner en práctica los conocimientos adquiridos. También se realizará la resolución de algunos prácticos en lápiz y papel.

Competencias que se promoverán en las prácticas de aula y laboratorio:

- Resolución de problemas.
- Capacidad de análisis y síntesis.
- Capacidad para aplicar la teoría a la práctica.
- Capacidad de generar nuevas ideas.

- Capacidad de razonamiento crítico.
- Capacidad de trabajo en equipo.
- Capacidad de abstracción, concreción, concisión, razonamiento, creatividad, síntesis y precisión.

Prácticos de aula:

Práctico 1: Representación computacional de conjuntos. Desarrollo de algoritmos. Familiarizarse con las distintas representaciones de conjuntos, diferenciando unas de otras para el posterior desarrollo de los algoritmos que permiten realizar la búsqueda de elementos de un conjunto en las distintas representaciones estudiadas.

Práctico 2: Notaciones asintóticas. Su utilización en la comparación de algoritmos. Decidir qué función de costo es más adecuada para la evaluación de un algoritmo, obtener la correspondiente notación asintótica y poder comparar entre los algoritmos que resuelven el mismo problema.

Práctico 3: Lista. Operadores para su administración. Almacenar una relación en los distintos tipos de listas para adquirir un conocimiento acabado de su manejo. Diseñar los operadores necesarios para administrar cada una de ellas (altas, bajas, inserciones, supresiones, evocaciones, memorizaciones).

Práctico 4: Pilas y Colas. Parva. Diseñar distintas implementaciones de pilas y colas al igual que los algoritmos que las administran. Implementar la parva y diseñar los algoritmos necesarios para manejarla. Implantación de las colas de prioridad utilizando estas estructuras. Análisis de otras Aplicaciones.

Práctico 5: Conceptos de teoría de grafos. Representación. Algoritmos. Afianzar las definiciones de grafos mediante la gráfica y análisis de distintos tipos de grafos. Realizar representaciones formales y computacionales. Diseñar algoritmos sobre grafos.

Práctico 6: Árboles como estructuras de información. Barridos. Árboles binarios. Almacenar una relación en los distintos tipos de árboles computacionales y diseñar los operadores que permiten su administración. Implementar los distintos barridos sobre árboles y algunas aplicaciones usando los mismos.

Práctico 7: Más listas: Lista de 2 niveles, Skip list. Almacenar relaciones en estas listas y diseñar los algoritmos que las administren, lo que permitirá un amplio conocimiento de las mismas, de sus ventajas y debilidades.

Práctico 8: Direccionamiento directo. Distribución pseudo-aleatoria de datos. Desarrollar ejercicios que impliquen la implementación de los distintos tipos de rebalse, permitiendo el entendimiento de ventajas y desventajas sobre otras estructuras. Analizar el direccionamiento directo, sus coincidencias y diferencias con los rebalses.

Práctico 9: Técnicas de diseño de algoritmos. Desarrollo de algoritmos de ordenamiento, de búsqueda, algoritmos sobre grafos y de diversos tipos aplicando las distintas técnicas. El conocimiento e implementación de algoritmos representativos para cada técnica de diseño permitirá adquirir habilidad y experiencia para utilizar la mejor técnica al momento de diseñar nuevos algoritmos al momento de su desempeño profesional.

Laboratorios:

En todos los laboratorios se estudian situaciones reales a resolver y cuya solución se deba implementar en algún lenguaje de programación, teniendo en cuenta los conceptos que gradualmente se vayan incorporando en las teorías y prácticos de aula. En cada caso el alumno deberá diseñar una solución adecuada al problema y deberá entregar el código fuente correspondiente, correctamente documentado, y un informe escrito que describa y justifique la solución planteada, los detalles de implementación, las elecciones realizadas, los costos involucrados, etc; además deberá entregar los resultados obtenidos con la ejecución de su trabajo y el análisis de los mismos. Mediante los proyectos de laboratorio se espera que el alumno realice una autoevaluación sobre lo aprendido y comprenda más profundamente los conceptos teóricos y prácticos involucrados. Los proyectos en sí agrupan laboratorios, con el fin de facilitar la comparación de las soluciones propuestas frente a problemas similares.

Laboratorio 1. Situación a resolver: Planteo de una situación real en la que se deban almacenar elementos de un conjunto y se

necesite realizar la búsqueda para responder si un elemento pertenece o no a ese conjunto. Metodología: El alumno debe revisar las estructuras de datos vistas hasta el momento, proponer la más adecuada, e implementarla para resolver el problema.

Laboratorio 2. Situación a resolver: Planteo de una situación real en la que se deba almacenar una relación y sea necesario realizar una evocación asociativa sobre la misma. Metodología: El alumno debe revisar las estructuras de datos vistas hasta el momento, proponer la más adecuada, e implementarla para resolver el problema.

Laboratorio 3. Situación a resolver: Planteo de una situación real en la que se deba almacenar una relación dinámica y sea necesario realizar todas las operaciones sobre la misma. Metodología: El alumno debe revisar las estructuras de datos vistas hasta el momento, proponer la más adecuada, e implementarla para resolver el problema, considerando las opciones más eficientes para cada operación.

Laboratorio 4. Situación a resolver: Planteo de una situación real en la que se deba almacenar una relación dinámica y sea necesario realizar todas las operaciones sobre la misma, sabiendo que el alta de elementos será una operación muy frecuente. Metodología: El alumno debe revisar las estructuras de datos vistas hasta el momento, proponer la más adecuada, e implementarla para resolver el problema, considerando las opciones más eficientes para cada operación.

Laboratorio 5. Situación a resolver: Planteo de una situación real en la que se deba almacenar una relación dinámica y sea necesario realizar todas las operaciones sobre la misma, sabiendo que el alta de elementos será una operación muy frecuente y que los mismos no serán insertados en orden. Metodología: El alumno debe revisar las estructuras de datos vistas hasta el momento, proponer la más adecuada, e implementarla para resolver el problema, considerando las opciones más eficientes para cada operación.

Laboratorio 6. Situación a resolver: Planteo de una situación real en la que se deba almacenar una relación dinámica y sea necesario realizar todas las operaciones sobre la misma. Metodología: El alumno debe revisar las estructuras de datos vistas hasta el momento, proponer la más adecuada, e implementar una de sus variantes para resolver el problema (rebalse abierto en alguna de sus variantes o rebalse separado), considerando las opciones más eficientes para cada operación.

Laboratorio 7. Situación a resolver: Planteo de una situación real en la que además de almacenar una relación dinámica se mantenga sobre los elementos de la relación una cola de prioridad y sea necesario realizar todas las operaciones sobre la relación y la cola de prioridad. Metodología: El alumno debe revisar las estructuras de datos vistas hasta el momento, proponer la más adecuada, e implementarlas para resolver el problema, considerando las opciones más eficientes para cada operación.

Laboratorio 8. Situación a resolver: Planteo de una situación real que deba modelarse mediante grafos dirigidos y en la que sea necesario resolver un problema concreto. Metodología: El alumno debe revisar las representaciones de grafos vistas, proponer la mejor para el caso planteado, diseñar el algoritmo que resuelva el problema considerado, de acuerdo a la técnica de diseño de algoritmos más adecuada, e implementarlo eficientemente.

Para una mejor comprensión de los conceptos involucrados y para facilitar al alumno la comparación y evaluación de las soluciones propuestas, los laboratorios se agruparán en proyectos, de acuerdo a la situación real presentada.

VIII - Régimen de Aprobación

Régimen para Alumnos Regulares:

- Asistencia a práctico: 70%
- Asistencia a teoría: 70%
- Presentación y aprobación de los prácticos de laboratorio propuestos. Para ello deben entregar el código fuente y un informe en el que se justifique la solución propuesta y se brinden detalles del trabajo realizado. Los códigos fuentes deben poder compilarse y resolver las situaciones planteadas. Cada práctico de laboratorio tendrá una posibilidad de recuperación para su aprobación.
- Aprobar el parcial o sus recuperaciones. Se toma un parcial, que tiene dos recuperaciones. En todos los casos la aprobación es con el 70%.

Modalidad de examen final:

El examen final podrá ser oral y/o escrito.

Examen Libre: No se admiten alumnos libres dado que los prácticos de laboratorio y aula se desarrollan de manera incremental desde comienzo de cuatrimestre, por consiguiente no es posible en un examen poder evaluar correctamente este proceso.

No es una materia promocional.

IX - Bibliografía Básica

- [1] INTRODUCTION TO ALGORITHMS, Autor: Cormen, Leiserson and Rivest, ISBN: 0262031418 -MIT-, The MIT Press, 3a edición, 2003, Ubicación en biblioteca: 519.254 C811.
- [2] FUNDAMENTOS DE ALGORITMIA, Autor: : Brassard, Gilles y Bratley, Paul, ISBN: 84-89660-00-X, Prentice Hall, 1a. edición, 2000, Ubicación en Biblioteca: 004.021.B823f.
- [3] COMPARED TO WHAT? : AN INTRODUCTION TO THE ANALYSIS OF ALGORITHMS, Autor: Rawlins, Gregory, ISBN: 071678243X, W. H. Freeman, 1991.
- [4] THE ART OF COMPUTER PROGRAMMING (VOL 1 Y 3), Autor: Knuth, Donald E., ISBN-10: 0201896834 y 0201896850, ISBN-13: 978-0201896831 y 978-0201896855, Addison-Wesley Professional, VOL 1: 3a edición, 1997, VOL 3: 2a edición, 1998, Ubicación en biblioteca: 681.3.06 K74.
- [5] ALGORITHMS IN C (PARTS 1-4), Autor: Sedgewick, Robert. 3a. Ed. ISBN 0-201-31452-5, Addison-Wesley Professional, 3a edición 1997, Ubicación en Biblioteca: 004.422.63 S448a3 I.
- [6] GRAPHS ET HYPERGRAPHS, Autor: Berge, C., ISBN-10: 0444876030, ISBN-13: 978-0444876034, North Holland, 1989, Ubicación en biblioteca: 519.28 B495.
- [7] AN INTRODUCTION TO THE ANALYSIS OF ALGORITHMS, Autor: Sedgewick, Robert and Flajolet, Philippe. ISBN: 020140009X, Addison-Wesley Professional, 1995, Ubicación en biblioteca: 004.021 S448.
- [8] ALGORITHMIC: THEORY AND PRACTICE, Autor: Brassard, Gilles and Bratley, Paul, Ubicación en biblioteca: 519.681 B823.
- [9] DATA STRUCTURES AND PROGRAM DESIGN IN C, Autores: Kruse, Robert L.; Leung, Bruce P.; Tondo, Clovis L., ISBN: 0-13-725649-3, Prentice Hall, 2a edición, 1996, Ubicación en biblioteca: 519.682.2681.3.06 K94.
- [10] DATA STRUCTURES AND THEIR ALGORITHMS, Autor: Lewis, Harry R., Denenberg, Larry, ISBN: 0-673-39736-X, Addison Wesley, 1991, Ubicación en biblioteca: 519.683.2 L674.
- [11] ESTRUCTURA DE DATOS y ALGORITMOS, Autor: Mark Weiss, ISBN: 0-201-62571-7, Addison Wesley Longman, 2000.
- [12] APUNTES DE LA CÁTEDRA: Durante el dictado se entregarán apuntes confeccionados por la cátedra sobre algunos de los temas. *Repaso de Conjuntos, Relaciones y Funciones. *Introducción a las Estructuras de Datos. *Descripción Informática de Conjuntos. *Pertenencia en Conjuntos Computacionales. *Operaciones en Conjuntos Computacionales. *Evaluación de Algoritmos. *Teoría de Grafos (Parte I). *Teoría de Grafos (Parte II). *Árboles Binarios Ordenados. *Costos de Búsqueda en Árboles Binarios de Búsqueda. * Árboles de Expresión. *Lista de 2 Niveles. *Distribución Pseudo-aleatoria de Datos. *Diseño de Funciones de Enumeración. *Deducción de algunos esfuerzos para distribución pseudo- aleatoria de datos. *Diseño de Funciones de pseudo-azar. *Estructuras de Datos Aleatorias: Skip Lists. *Árboles Digitales: Trie y Patricia.

X - Bibliografía Complementaria

- [1] COMPUTER ALGORITHMS: INTRODUCTION TO DESIGN AND ANALYSIS, Autor: Baase, Sara, ISBN: 0-201-06035-3, Addison Wesley, 3a edición, 1999, Ubicación en biblioteca: 519.682.4 B111c2
- [2] DATA STRUCTURES AND ALGORITHMS, Autor: Aho, Hopcroft, Ullman, ISBN: 0-201-00023-7, Addison Wesley, 1983, Ubicación en biblioteca: 519.683.2 A286
- [3] CONCRETE MATHEMATICS, Autor: Graham, Ronald L., Knuth, Donald E. , Patashnik, Oren, ISBN: 0-201-55802-5, Addison-Wesley Professional, 2 edición, 1994, Ubicación en biblioteca: 511.333 G741c2
- [4] THE DESIGN AND ANALYSIS OF COMPUTER ALGORITHMS, Autor: Aho, Hopcroft, Ullman, ISBN: 0-201-00029-6, Addison-Wesley, 1974, Ubicación en biblioteca: 519.683.2 A286
- [5] DATA STRUCTURES & PROGRAM DESIGN, Autor: Kruse, Robert, ISBN: 0-132-08182-2, Prentice-Hall, 2a edición,

1994, Ubicación en biblioteca: 681.3.06 K94D2

[6] DATA STRUCTURES TECHNIQUES, Autor: Standish, T., ISBN: 0-201-07256-4, Addison-Wesley, 1980, Ubicación en biblioteca: 681.3.0651.S785

[7] COMPUTER ALGORITHMS: KEY SEARCH STRATEGIES, IEEE Computer Society Technology Series, ISBN: 0-818-69123-9, IEEE Computer Society, 1990, Ubicación en biblioteca: 519.681.5519.878681.3.06 A638

[8] MATHEMATICS FOR THE ANALYSIS OF ALGORITHMS, Autor: Greene, Daniel , Knuth, Donald, ISBN: 0-8176-3515-7 (Birkhäuser), ISBN: 3-7643-3515-7 (Boston-Basel-Berlín), Birkhäuser Boston; 3a edición Rev., 2007.

[9] PUGH, WILLIAM: Skip Lists: A Probabilistic Alternative to Balanced Trees. Communications of the ACM, 33, 1990,pág. 668-676.

[10] LEWIS, T y COOK, C: Hashing for Dynamic and Static Internal Tables. IEEE Comp. Oct.1988.

[11] LARSON, P: Dynamic Hash Tables. C. ACM, Vol 31 N0 4, Abril 1988.

[12] SLEATOR D. D. y TARJAN R. E.: Self-adjusting binary search trees Journal of ACM, ISSN 0004-5411, Vol 32, Nro.3, pág. 652-686, Julio 1985.

XI - Resumen de Objetivos

Al finalizar el curso se pretende que el alumno sea capaz de:

- Manejar los conceptos que involucran el diseño de estructuras de datos.
- Conocer los principales algoritmos y estructuras de datos y el análisis de su desempeño.
- Analizar y diseñar algoritmos.
- Frente a una aplicación o problema particular, poder brindar una solución eficiente utilizando los conceptos vistos sobre diseño de estructuras de datos y algoritmos, introducir herramientas de diseño de algoritmos y la ingeniería algorítmica como selección de las estructuras de datos y de las técnicas algorítmicas más adecuadas y además utilizar el análisis de los algoritmos para evaluar y justificar la eficiencia de la solución elegida.

XII - Resumen del Programa

1. Explicitación del objetivo de la materia: Las estructuras. Dato e información. Datos de un problema de estructuras de datos. Algoritmos.

2. Evaluación de algoritmos: Planteo. Problema. Medidas en tiempo y espacio. Familias de problemas. Notaciones asintóticas. Clases de Complejidad.

3. Listas, Pilas y Colas: Listas vinculadas y secuenciales. Búsqueda binaria. Lista Invertida. Lista de 2 niveles. Pilas y colas. Skip Lists. Operaciones sobre cada tipo de estructura. Aplicaciones.

4. Propiedades, demostraciones y representación de grafos: Definiciones. Propiedades. Representaciones básicas de grafos. Algoritmos sobre grafos.

5. Árboles computacionales: Definición. Representación computacional. Árboles Binarios de Búsqueda. Árboles Balanceados en altura. Parva. Operaciones básicas sobre cada tipo de árbol. Aplicaciones.

6. Distribución pseudo-aleatoria de datos: Motivación. El problema del rebalse. Distintas propuestas para su manejo. Direccionamiento Directo. Aplicaciones.

7. Técnicas de diseño de algoritmos: Técnicas ávidas, programación dinámica, dividir para vencer. Ejemplos de aplicación de las distintas técnicas.

XIII - Imprevistos

XIV - Otros

--