

PROGRAMA DE CURSO ALGORITMOS COMBINATORIALES

A. Antecedentes generales del curso:

Departamento	DIM					
Nombre del curso	Algoritmos combinatoriales	Código	MA3705	Créditos	6	
Nombre del curso en inglés	<i>Combinatorial Algorithms</i>					
Horas semanales	Docencia	3	Auxiliares	1,5	Trabajo personal	5,5
Carácter del curso	Obligatorio	X		Electivo		
Requisitos	MA3711: Optimización matemática					

B. Propósito del curso:

El curso tiene como propósito que los y las estudiantes conozcan los algoritmos clásicos para problemas combinatoriales, principios generales de diseño de algoritmos y la noción de eficiencia computacional. También que sean capaces de reconocer la complejidad computacional de un problema de optimización discreta, y presentar técnicas para aproximar problemas NP-duros. Además, que comprendan la relevancia, tanto en el diseño como en el análisis de algoritmos combinatoriales, de la formulación primal-dual de programación lineal.

El curso tributa a las siguientes competencias específicas (CE):

CE1: Interpretar y utilizar el lenguaje formal matemático, para analizar y verificar la veracidad de afirmaciones matemáticas.

CE2: Calcular y manipular objetos matemáticos y herramientas conceptuales de diversas áreas de las matemáticas, tales como análisis, simulación numérica, ecuaciones diferenciales, matemáticas discretas, optimización, probabilidades y estadísticas, entre otras, para la resolución de problemas.

CE3: Modelar matemáticamente problemas de diferentes áreas en situaciones simples, es decir, traducir la realidad a una estructura matemática de forma tal que se facilite su análisis.

CE5: Concebir, diseñar y evaluar desarrollos científico-tecnológicos para resolver problemas en el ámbito de las ciencias de la ingeniería.

El curso tributa a las siguientes competencias genéricas (CG):

CG1: Comunicación académica y profesional. Comunicar en español de forma estratégica, clara y eficaz, tanto en modalidad oral como escrita, puntos de vista, propuestas de proyectos y resultados de investigación fundamentados, en situaciones de comunicación compleja, en

ambientes sociales, académicos y profesionales.

CG3: Compromiso ético. Actuar de manera responsable y honesta, dando cuenta en forma crítica de sus propias acciones y sus consecuencias, en el marco del respeto hacia la dignidad de las personas y el cuidado del medio social, cultural y natural.

Por su naturaleza, los resultados de aprendizaje RA6 y RA7 son parte de cada una de las unidades y su validación se hará en las actividades de evaluación.

C. Resultados de aprendizaje:

Competencias específicas	Resultados de aprendizaje
CE2	RA1: Aplica los algoritmos analizados para encontrar soluciones en ejemplos concretos.
CE3	RA2: Modela una situación simplificada de la realidad como uno de los problemas clásicos presentados, considerando las ventajas de optar por uno o por otro.
CE1-CE2	RA3: Adapta argumentos teóricos de los temas analizados a variaciones de problemas discutidos para justificar propiedades matemáticas de los modelos que se construyen.
CE2-CE5	RA4: Analiza la estabilidad de los algoritmos ante modificaciones simples en su diseño, determinando el efecto de estas en sus tiempos teóricos.
CE2-CE5	RA5: Identifica la dificultad probable de un problema y los algoritmos apropiados para su solución exacta o aproximada.
Competencias genéricas	Resultados de aprendizaje
CG1	RA6: Argumenta por escrito, tanto en controles, exámenes o tareas asociadas, los resultados obtenidos en la solución de problemas, con especial cuidado en la claridad y precisión en el uso de los términos matemáticos.

CG3	RA7: Realiza las actividades programadas, cumpliendo con sus requerimientos, plazos y de manera honesta, en particular, sin plagiar trabajos en tareas o informes, ni copiar en evaluaciones.
-----	---

Por su naturaleza, los resultados de aprendizaje RA4 y RA5 son parte de cada una de las unidades y su validación se hará en las actividades de evaluación.

D. Unidades temáticas:

Resumen de unidades temáticas

Unidad	Nombre de la unidad	Duración
1	Medidas de eficiencia computacional	0.5
2	Problema de árbol generador de peso mínimo.	2.0
3	Problema de camino más corto.	1.5
4	Problema de cuplajes y cubrimientos por vértices en grafos bipartitos.	1.5
5	Problema de flujos y cortes	2.5
6	Problema de cuplaje: caso general	1.5
7	Separación y optimización	2.0
8	Intratabilidad	1.0
9	Algoritmos de aproximación	2.5
TOTAL		15

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA3,5	Medidas de eficiencia computacional	0.5 semanas
Contenidos		Indicador de logro	
1.1. Random Access Machine (RAM) y medida de complejidad de tiempo.		El/la estudiante: 1. Determine la complejidad de tiempo de algoritmos simples, usando el concepto de RAM	
Bibliografía de la unidad		[PS82, Cap. 2]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
--------	-------------------	---------------------	---------------------

2	RA1,2,3,4	El problema del árbol generador de peso mínimo	2 semanas
Contenidos		Indicador de logro	
<p>2.1. Algoritmo primal-dual para el problema del árbol generador de peso mínimo.</p> <p>2.2. Algoritmos de Kruskal, Prim y formulación primal-dual. Ilustración de éstos como una heurística de optimización basada en la estrategia glotona.</p> <p>2.3. Matroides y submodularidad de la función de rango. Optimalidad del algoritmo glotón para matroides y caracterización de matroides a través del algoritmo glotón.</p>		<p>El/la estudiante:</p> <ol style="list-style-type: none"> 1. Modela una situación simplificada de la realidad como un problema del árbol generador de peso mínimo. 2. Aplica los algoritmos para el problema del árbol generador de peso mínimo en ejemplos concretos. 3. Analiza variantes del problema del árbol generador de peso mínimo. 4. Evalúa el efecto de cambios en los algoritmos de Kruskal y Prim. 5. Argumenta acerca de la validez de nuevas propiedades de matroides. 	
Bibliografía de la unidad		[CLR09, Cap. 15, 24 y 25], [KT06, Cap. 6], [KV10, Cap. 7], [CCPS98, Cap. 2 x2].	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
3	RA1,2,3,5	El problema del camino más corto	1.5 semanas
Contenidos		Indicador de logro	
3.1. Algoritmo de Bellman visto como una ilustración de un método genérico de diseño de algoritmos basado en programación dinámica. 3.2. Algoritmos de Dijkstra, Bellman-Ford y Floyd-Warshall.		El/la estudiante: <ol style="list-style-type: none"> 1. Modela una situación simplificada de la realidad como un problema del camino más corto. 2. Analiza variantes del problema del camino más corto. 3. Determina aplicabilidad de los algoritmos de la unidad en base a las propiedades de los ejemplos. 4. Identifica las limitaciones para el uso de los algoritmos de la unidad. 	
Bibliografía de la unidad		[CLR09, Cap. 15, 24 y 25], [KT06, Cap.6], [KV10, Cap. 7], [CCPS98, Cap. 2 x2].	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
4	RA1,2,3,5	Problemas de cuplaje y cubrimiento por vértices en grafos bipartitos	1.5 semanas
Contenidos		Indicador de logro	
4.1. Cuplajes de máxima cardinalidad en grafos bipartitos. Caminos y árboles alternantes. Cubrimiento por vértices. Teorema de König. Cuplajes perfectos. Teorema de Hall. Teorema de Dilworth en órdenes parciales. 4.2. Cuplajes de peso máximo en grafos bipartitos y cuplajes perfectos de peso mínimo: El problema de asignación. Método Húngaro Primal-Dual para el problema de asignación. 4.3. Teorema de Birkhoff Von Neumann.		El/la estudiante: <ol style="list-style-type: none"> 1. Modela una situación simplificada de la realidad como un problema de cuplaje en grafos bipartitos. 2. Aplica Teoremas de König, Hall y Dilworth para justificar propiedades matemáticas de modelos. 3. Calcula soluciones a problemas de cuplaje usando los algoritmos de la unidad. 4. Selecciona qué algoritmo de la unidad es más indicado de usar en situaciones específicas. 	
Bibliografía de la unidad		[KT06, Cap. 4]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
5	RA1,2,3,4	Problemas de flujos y cortes	2.5 semanas
Contenidos		Indicador de logro	
<p>5.1. Teorema de Menger. Teorema del Flujo máximo y corte mínimo. Método primal-dual para problemas de flujo. Algoritmo de Ford-Fulkerson: el problema de la finitud del algoritmo y el Teorema de integralidad.</p> <p>5.2. Algunas mejoras del algoritmo de Ford-Fulkerson; los algoritmos de Edmonds y Karp. Sub-modularidad de la función de capacidad de cortes.</p> <p>5.3. Árboles de Gomory-Hu. Problema de cortes de multiterminales.</p>		<p>El/la estudiante:</p> <ol style="list-style-type: none"> 1. Modela una situación simplificada de la realidad como un problema de flujo. 2. Identifica las limitaciones para el uso de los algoritmos de la unidad. 3. Analiza cambios en los algoritmos que reducen sus tiempos teóricos en la búsqueda de las soluciones. 4. Aplica Teoremas de Menger y Flujo máximo-corte mínimo para validar propiedades de modelos. 5. Argumenta la validez del uso de uno de los algoritmos de la unidad para resolver un problema específico. 	
Bibliografía de la unidad		[CLR09, Cap. 26], [KV10, Cap. 8], [CCPS98, Cap. 3].	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
6	RA1,2,3	El problema del cuplaje en grafos generales	1.5 semanas
Contenidos		Indicador de logro	
<p>6.1. Cuplajes de cardinalidad máxima en grafos generales. Algoritmo de Blossom. Fórmula de Tutte-Berge.</p> <p>6.2. Integralidad del polítopo de cuplajes.</p>		<p>El/la estudiante:</p> <ol style="list-style-type: none"> 1. Modela una situación simplificada de la realidad como un problema de cuplaje. 2. Explica los argumentos que justifican la validez de los algoritmos de la unidad. 3. Aplica algoritmo de Blossom para justificar la validez de propiedades de grafos generales. 4. Asocia las soluciones al problema de cuplaje y los vértices de un polítopo. 	
Bibliografía de la unidad		[PS82, Caps.10,11], [CCPS98, Cap. 5].	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
7	RA4,5	Separación y optimización	2 semanas
Contenidos		Indicador de logro	
7.1. Método elipsoidal. Equivalencia entre el problema de separación. 7.2. Eficiente para poliedros y optimización eficiente de programas lineales. 7.3. Aplicación: Separación y optimización sobre el polítopo de cuplajes.		El/la estudiante: 1. Explica los argumentos que justifican la validez del método elipsoidal para resolver problemas lineales. 2. Construye argumentos para justificar la relación entre separación y optimización sobre el polítopo de cuplaje.	
Bibliografía de la unidad		[KT06, Cap. 4]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
8	RA4,5	Intratabilidad	1 semana
Contenidos		Indicador de logro	
8.1. Certificados de pertenencia. 8.2. Reducciones a tiempo polinomial. 8.3. Enunciar NP-completitud de SAT. 8.4. Ejemplo de reducciones.		El/la estudiante: 1. Identifica qué problemas vistos en las unidades anteriores admiten certificados de pertenencia. 2. Explica argumentos que justifican que SAT es un problema NP-completo. 3. Construye reducciones polinomiales entre problemas combinatoriales.	
Bibliografía de la unidad		[KT06, Cap. 8], [CLR09, Cap. 34].	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
9	RA4,5	<i>Algoritmos de aproximación</i>	2.5 semanas
Contenidos		Indicador de logro	
9.1. Definición de algoritmos de aproximación, esquemas de aproximación polinomial y demostraciones de inaproximabilidad. 9.2. Técnicas de aproximación. Algoritmos glotones, método aleatorio y derandomización vía esperanzas condicionales. 9.3. Algoritmos basados en programación lineal: redondeo, método		El/la estudiante: 1. Identifica la dificultad probable de un problema. 2. Evalúa las técnicas de aproximación en problemas específicos. 3. Explica los argumentos de inaproximabilidad en variaciones a problemas discutidos. 4. Construye algoritmos de aproximación para variaciones de los problemas discutidos.	

primal-dual y gaps de integralidad. 9.4. Esquemas de aproximación vía redondeo y enumeración.	
Bibliografía de la unidad	[Wo98, Cap. 13], [NW88, Cap. II.6].

E. Estrategias de enseñanza - aprendizaje:

El curso considera las siguientes estrategias de enseñanza-aprendizaje:

- Se realizarán clases presenciales lectivas.

F. Estrategias de evaluación:

Al inicio de cada semestre, el cuerpo académico informará oficialmente sobre la cantidad y tipo de evaluaciones, así como de sus ponderaciones. También anunciará si una inasistencia justificada se recupera mediante una evaluación adicional en las semanas siguientes a la evaluación original o al final del semestre, dependiendo del porcentaje de asistencia del estudiantado a la misma, o por la nota del examen.

Tradicionalmente hay distintas instancias de evaluación tales como:

- Evaluaciones parciales (controles, tareas, trabajo en clases, entre otros). Con un máximo de 3 controles por semestre.
- Examen final.

La ponderación de cada evaluación respetará los reglamentos de la Escuela. En cada uno de estos controles y examen final se evaluará la capacidad del estudiante para escribir proposiciones abstractas de manera clara y precisa.

Esta evaluación se realiza de manera integral en la revisión de las evaluaciones y puede afectar un porcentaje de la calificación de cada una de ellas.

Según el reglamento de estudios de la FCFM, el profesor tiene la facultad de realizar un examen oral a un estudiante. Esta instancia podrá darse, por ejemplo, cuando el alumno presente inasistencias reiteradas a los controles. De ser examinado en ambas formas (escrita y oral), recibirá calificaciones parciales separadas, las que se promediarán aritméticamente para dar la calificación del examen.

G. Recursos bibliográficos:

Bibliografía obligatoria:

- [1] [BY98] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge Press, 1998.
- [2] [CCPS98] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, y A. Schrijver, Combinatorial Optimization, John Wiley & Sons, Series in Discrete Mathematics and Optimization, 1998.
- [3] [CLR09] T.H. Cormen, C.E. Leiserson, R.L. Rivest, y C. Stein. Introduction to Algorithms, MIT Press, tercera edición, 2009.

- [4] [GLS93] M. Grötschel, L. Lovasz, y A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Series in Algorithms and Combinatorics, No. 2, Springer Verlag, second edition, 1993.
- [5] [Go1-91] M. Goemans, apuntes del curso Advanced Algorithms, MIT, 1991.
- [6] [Go2-94] M. Goemans, On Line Algorithms, apuntes del curso Advanced Algorithms, MIT, septiembre 1994.
- [7] [KT06] J. Kleinberg y E. Tardos, Algorithm Design, Addison Wesley, 2006.
- [8] [KV10] B. Korte, y J. Vygen, Combinatorial Optimization: Theory and Algorithms, Series in Algorithms and Combinatorics, Volume 21. Cuarta edición, Springer, 2010.
- [9] [Lee04] J. Lee, A First Course in Combinatorial Optimization, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2004.
- [10] [PS82] C.H. Papadimitriou, Computational Complexity, Addison Wesley, 1994.
- [11] [PS82] C.H. Papadimitriou y K. Steiglitz, Combinatorial Optimization, Algorithms and Complexity, Prentice Hall, 1982.
- [12] [Sch03] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency Springer Verlag, Series in Algorithms and Combinatorics, Vol. 24, 2003.
- [13] [Vaz01] V. V. Vazirani, Approximation algorithms. Springer Verlag, 2001.
- [14] [WS11] D. P. Williamson y D. B. Shmoys, The Design of Approximation Algorithms. Cambridge Press, 2011.

H. Datos generales sobre elaboración y vigencia del programa de curso:

Vigencia desde:	Primavera 2013
Elaborado por:	Marcos Kiwi y José Soto
Validado por:	Jefe Docente (2024)