FOLIO N°
114
Fa.M.A.F.

UNC Universidad Nacional de Córdoba
FAMAF Facultad de Matemática, Astronomía, Física y Computación

EXP-UNC 23224/2016 Res. CD N° 141/2016

PROGRAMA DE ASIGNATURA	
ASIGNATURA: Matemática Discreta II	**AÑO:** 2016
CARACTER: Obligatoria	**UBICACIÓN EN LA CARRERA:** 3° año 1° cuatrimestre
CARRERA: Licenciatura en Ciencias de la Computación	
REGIMEN: Cuatrimestral	**CARGA HORARIA:** 120 horas

FUNDAMENTACIÓN Y OBJETIVOS

Objetivos: Que los alumnos adquieran:

1) Experiencia en el desarrollo de algoritmos complejos y en el análisis y prueba de sus complejidades. En este ultimo punto, que aprendar a leer y escribir sobre conceptos del area.

2) Comprensión de conceptos de flujos maximales, matchings, complejidad computacional, P-NP, códigos de corrección de errores, importancia de algoritmos polinomiales, nociones básicas de algoritmos genéticos.

CONTENIDO

Coloreo

Repaso de la noción de grafo. Notaciones. Coloreo de Grafos. Número cromático. Algoritmo de fuerza bruta. Problema k-Color. Definición de bipartito. Repaso de BFS.

Propiedad: un grafo es bipartito si y solo si no tiene ciclo impares. Algoritmo polinomial para 2-COLOR. Algoritmo Greedy de Coloreo. Ejemplo de aplicación. Ejemplo de que no funciona. Propiedad: $\chi(G) \leq$ Delta(G)+ 1.

Ejemplos de que se alcanza la cota. Algoritmo de Brooks. Teorema de Brooks (sin prueba general. Prueba solo para el caso particular de grafos no regulares) Algoritmo que implementa Brooks en el caso de grafos no regulares.

Algoritmos de coloreo WelshPowell,DSATUR, RLF y Greedy iterado.

Flujos

Grafos Dirigidos. Ejemplos. Networks. Flujos. Propiedad: out(s) – in(s) = in(t) – out(t). Valor de un flujo. Definición de corte y capacidad de un corte. Propiedad: v(f) = f(S,\overline{S}) - f(\overline{S},S). Corolario: el valor de cualquier flujo es menor o igual que la capacidad de cualquier corte. Corolario: Si el valor de un flujo es igual a la capacidad de un corte entonces el flujo es maximal y el corte es minimal. Algoritmo "Greedy" para encontrar un flujo maximal. Ejemplo donde funciona. Ejemplo donde falla. Como modificar el algoritmo para que funcione. El Teorema del flujo maximal-Corte minimal (Max-Flow-Min-Cut Theorem). Algoritmo de Ford-Fulkerson. Ejemplos de aplicación del algoritmo de Ford-Fulkerson. Debilidades del algoritmo de Ford-Fulkerson: ejemplo donde la complejidad no depende del número vértices o lados. Ejemplo donde el algoritmo no termina. Refinamientos: Algoritmos fuertemente polinomiales: Algoritmo de Edmonds-Karp. Complejidad. Algoritmo de Dinic. Complejidad. Algoritmos de pre-flow/push: algoritmo "wave" de Tarjan. Complejidad.

Matchings

Matchings en grafos bipartitos. Matchings perfectos y Matchings completos. Ejemplos. Algoritmos para encontrar matchings como aplicación de los algoritmos para encontrar flujos maximales. Modificaciones. Uso de matrices. Definición de Gamma(S). Condición de Hall. Teorema de Hall. Teorema del matrimonio. (Todo grafo bipartito regular tiene un matching perfecto). Problemas de Matchings Óptimos en grafos bipartitos con pesos. Resolución del "bottleneck problem": problema del asignamiento óptimo cuando se desea minimizar el máximo (o maximizar el mínimo) de los pesos.

Resolución del problema del asignamiento óptimo cuando se desea minimizar (o maximizar) la suma de los pesos: Algoritmo Húngaro. Codificación de complejidad O(n) del algoritmo Húngaro.

Tópicos de Inteligencia Artificial
Algoritmos de búsqueda. Hill Climbing. Simulated Annealing.
Algoritmos genéticos: Codificación del problema. Fitness. Reproducción de Población. Terminación.
Elementos de la Reprodución: Selección, Crossover, Mutación, Reemplazo.
Algunas posibilidades de Selección para Reproducción: Roulette, SUS, Rank-based selection.
Algunas posibilidades de Crossover. Single point, double or multiple points.
Algunas posibilidades de Mutación. Algunas posibilidades de Reemplazo: Ambos progenitores, Progenitor más débil, Individuos más débiles, Randmo. Ventajas y desventajas.

Intrpoduccion a Códigos
Definiciones básicas. Distancia de Hamming. Detección de errores. Corrección de errores. Ejemplos de códigos. Chequeo de paridad. Códigos de repetición. Cota de Hamming. Códigos lineales. Propiedad: C lineal entonces delta(C) es igual al mínimo peso no nulo. Matrices Generadoras. Códigos lineales como espacios fuilas de una matriz. Códigos lineales como núcleos de matrices. Matrices de chequeo. Equivalencias entre matrices generadoras y de chequeo. Propiedad: todo código lineal tiene un matriz de chequeo. Proposición: Si en la matriz de chequeo no hay columnas repetidas ni nulas entonces el código correspondiente corrige al menos un error. Algoritmo para corregir un error. Códigos de Hamming. Códigos perfectos. Propiedad: Hamming es perfecto. Propiedad: delta(C) es igual al menor número de columnas linealmente dependientes de una matriz de chequeo. Singleton Bound. Códigos MDS.

Códigos Cíclicos
Rotación de una palabra. Códigos cíclicos. Códigos cíclicos mirados como polinomios. Propiedad: todo código lineal (sobre Z_2) tiene un único polinomio no nulo de menor grado. (arriba Z_2 denota los enteros modulo dos). Polinomio generador de un código cíclico. Propiedades del polinomio generador. Uso del polinomio generador para codificación: dos métodos. Matrices generadoras asociadas a los dos métodos. Obtención en forma directa a partir del polinomio generador de una matriz de chequeo con la identidad a izquierda. Polinomio chequeador. Corrección de errores: error trapping.

Códigos de Reed-Solomón
Cuerpos finitos. Definiciones y construcción. Propiedades básicas de los cuerpos finitos. Teorema del elemento primitivo. (dependiendo del tiempo la prueba puede ser omitida). Códigos de Reed-Solomon. Teorema: los códigos de Reed-Solomon son MDS.

P y NP
La clase P. La clase NP. Ejemplos. El problema SAT. El problema k-COLOR. Ejemplo: 2-COLOR es P. Reducción polinomial. Reducción de 3-COLOR a SAT. Las clases de problemas NP-hard y NP-completo. Teorema de Cook (sin prueba): SAT es NP-completo. Teorema: 3-SAT es NP-completo. Teorema: 3-COLOR es NP-completo. Dependiendo del tiempo: 2-SAT está en P. Horn-SAT está en P.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

1) Data Structures and Network Algorithms. R.E. Tarjan. 1983, Society for Industrial and Applied Mathematics.
2) Computers and Intractability: A Guide to the Theory of NP-completeness. Garey and Johnson, 1979, Bell Telephone Laboratories.
3) .pdfs de algunos temas en página de Daniel Penazzi.
4) Matemática Discreta. N. Biggs, 1989.
5) Network Flows: Theory, Algorithms and Applications. Ahoja-Magnani-Orlin, 1993, Prentice Hall.

BIBLIOGRAFÍA COMPLEMENTARIA

1) Applied Combinatorics. Roberts, 1989. Prentice-Hall.
2)Applied Combinatorics. A. Tucker, 2nd Ed.,1984.
3) Combinatorial Optimization: Algorithms and Complexity. Papadimitriou-Steiglitz, 1998, Dover Publications.

EVALUACIÓN

FORMAS DE EVALUACIÓN

2 parciales (cada uno con un recuperatorio).
1 proyecto de programacion.
1 final.

REGULARIDAD

1. cumplir un mínimo de 70% de asistencia a clases teóricas, prácticas, o de laboratorio.
2. aprobar al menos dos evaluaciones parciales o sus correspondientes recuperatorios.
3. aprobar el Trabajo de Laboratorio.

PROMOCIÓN

No hay régimen de promoción en le cursado de la materia.