PROGRAMA DE ASIGNATURA	
ASIGNATURA: Matemática Discreta II	**AÑO:** 2025
CARACTER: Obligatoria	**UBICACIÓN EN LA CARRERA:** 3° año 1° cuatrimestre
CARRERA: Licenciatura en Ciencias de la Computación	
REGIMEN: Cuatrimestral	**CARGA HORARIA:** 120 horas

ASIGNATURA: Matemática Discreta II	**AÑO:** 2025
CARACTER: Obligatoria	**UBICACIÓN EN LA CARRERA:** 4° año 1° cuatrimestre
CARRERA: Licenciatura en Matemática Aplicada	
REGIMEN: Cuatrimestral	**CARGA HORARIA:** 120 Horas.

FUNDAMENTACIÓN Y OBJETIVOS

Esta materia aborda temas de Matemática Discreta, Teoría de Códigos de Corrección de Errores, Teoría de Complejidad rudimentos de inteligencia artificial.

La parte principal de la materia es el estudio de algoritmos sobre grafos y networks, y especialmente el análisis de la corrección y las complejidades de los mismos.

El objetivo de esta parte es que los/as estudiantes comprendan que en muchas aplicaciones no basta dar un algoritmo sino que hay que demostrar su correctitud, dar una cota de su complejidad y demostrarla. Convergen el desarrollo de habilidades de algoritmia y matemática.

Además de esta parte central la materia brinda formación general sobre códigos de corrección de errores, el problema P-NP, especialmente pertinente al tratarse problemas, en la primera parte, para los que no se conocen algoritmos polinomiales. Por último, se abordan problemas de inteligencia artificial y se proponen algoritmos genéticos para ejemplificar posibles cursos de acción cuando no se cuenta con algoritmos polinomiales. Dependiendo del tiempo, tambien se podra dar algunos conceptos basicos de Machine Learning

CONTENIDO

1. Coloreo de Grafos

Repaso de la noción de grafo. Notaciones. Coloreo de Grafos. Numero cromático. Algoritmo de fuerza bruta. Problema k-Color. Definición de bipartito. Conectitud. Componentes conexas. Repaso de BFS y DFS. Algoritmo polinomial para determinar biparticidad. Propiedad: un grafo es bipartito si y solo si no tiene ciclo impares. Algoritmo Greedy de Coloreo. Ejemplos de aplicación. Ejemplo de que no siempre Greedy devuelve el número cromático.
Ejemplo de que tan mal puede dar.
Teorema importante (central para el proyecto):
Sea G=(V,E) un grafo cuyos vértices están coloreados con un coloreo propio c con r colores {0,1,...,r-1}. Sea P una permutación de los números 0,1,...,r-1. Sea V[i]={x en V tal que c(x)=i}, i=0,1,...,r-1. Ordenemos los vértices poniendo primero los vértices de V[P(0)], luego los de V[P(1)],etc, hasta $V[P(r-1)]. Entonces Greedy en ese orden coloreará G con r colores o menos.
Propiedad: El número cromático es menor o igual que Delta +1. Ejemplos donde se alcanza la cota.
Teorema de Brooks. (prueba solo para el caso no regular).
Teorema de los cinco colores.

2. Fundamentos de inteligencia artificial

Algoritmos de Búsqueda. Hill Climbing. Simulated Annealing. Algoritmos Genéticos: Codificación del problema. Fitness. Reproducción de Población. Terminación.
Selección, Crossover, Mutación, Reemplazo. Algunas posibilidades de Mutación. Algunos posibilidades de Crossover. Single point, double, multiple points o mascara. Crossover en el caso de permutation based codifications: crossover básico, Partial Mixing Crossover y Ciclico.
Algunas posibilidades de Selección: Ruleta, SUS, Rank-based selection. Sigma based selection.
Otras posibilidades de estructura: catástrofes e islas. con migraciones.

Machine Learning Algorithms (dependiendo del tiempo)

3. Flujos Maximales

Grafos Dirigidos. Ejemplos. Networks (redes). Flujos sobre redes. Valor de un flujo. Flujos maximales. Diversos ejemplos. Algoritmo Greedy para encontrar flujo maximal. Ejemplo donde no necesariamente encuentra flujo maximal. Definición de corte y capacidad de un corte. Caminos aumentantes de Ford-Fulkerson. Algoritmo de Ford-Fulkerson.
Propiedad: Al aumentar el flujo a lo largo de un camino aumentante de Ford-Fulkerson lo que se obtiene sigue siendo flujo.
Max Flow Min Cut Theorem:
a) El valor de todo flujo es menor o igual que la capacidad de todo corte.
b) Si f es un flujo, las siguientes afirmaciones son equivalentes:
1) f es maximal.
2) Existe un corte S tal que v(f)=cap(S).
3) No existen f-caminos aumentantes.
Ejemplos de aplicación del algoritmo de Ford-Fulkerson. Debilidades del algoritmo de Ford-Fulkerson:Ejemplo donde la complejidad no depende del numero de vértices o lados.
Ejemplo donde el algoritmo no termina.
Refinamientos: Algoritmos fuertemente polinomiales: Algoritmo de Edmonds-Karp. Complejidad. Algoritmo de Dinic (o Dinitz). Complejidad de sus 2 versiones. Algoritmos de pre-flow/push: algoritmo ``wave'' de Tarjan. Complejidad.

4. Matchings

Matchings en grafos bipartitos, Matchings perfectos y Matchings completos. Ejemplos.
Algoritmo 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.
Resolucion 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 al cubo) del algoritmo Húngaro.

5. Códigos de corrección de errores.

Códigos de corrección de errores. Definiciones básicas. Distancia de Hamming. Detección y Corrección de errores. Ejemplos de códigos. Chequeo de paridad. Códigos de repetición. Cota de Hamming.
Códigos Lineales. Propiedad: Si C lineal entonces delta(C) es igual al mínimo peso no nulo.
Matrices Generadoras. Códigos lineales como espacios filas de una matriz.
Códigos lineales como núcleos de matrices. Matrices de chequeo.
Equivalencia entre matrices generadoras y de chequeo. Propiedad: todo código lineal tiene una 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. Generalización de esta propiedad a

corrección de más errores: (Teorema:) Si H es una matriz de chequeo de C, entonces delta(C)=Min{j: existe un conjunto de j columnas linealmente dependientes de H}.
Algoritmo para corregir un error. Códigos de Hamming. Códigos perfectos. Propiedad: Hamming es perfecto. Singleton Bound. Códigos MDS
Códigos Cíclicos. Rotación de una palabra. Códigos cíclicos. Códigos cíclicos mirados como polinomios. Propiedad: todo código lineal binario tiene un único polinomio no nulo de menor grado. Definición de 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 ReedSolomon.

6. P-NP.
La clases P y NP. Ejemplos. El problema SAT. El problema k-COLOR. Reducción polinomial. Las clases de problemas NP-hard y NP-completo.
Teorema de Cook: SAT es NP-completo. Teorema: 3-SAT es NP-completo. Teorema: 3-COLOR es NP-completo.
Teorema: Matrimonio trisexual es NP-completo.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA
- Matemática Discreta. N. Biggs,1989
- Applied Combinatorics. Roberts, 1989, Prentice-Hall.
- Data Structures and Network Algorithms. R.E. Tarjan, 1983, Society for Industrial and Applied Mathematics.
- Computers and Intractability: A Guide to the Theory of NP-completeness. Garey and Johnson, 1979, Bell Telephone Laboratories.
- Apuntes del docente de la materia, Daniel Penazzi.
- Combinatorial Optimization: Algorithms and Complexity. Papadimitriou-Steiglitz, 1998, Dover Publications.

BIBLIOGRAFÍA COMPLEMENTARIA
- Applied Combinatorics. A. Tucker, 2nd Ed., 1984
- Network Flows: Theory, Algorithms and Applications.
Ahoja-Magnani-Orlin, 1993, Prentice-Hall

EVALUACIÓN

FORMAS DE EVALUACIÓN
Habrá dos evaluaciones parciales con sus respectivos recuperatorios.

La evaluación final constará de dos partes: una parte teórica, sobre demostraciones de resultados teóricos y una parte práctica sobre resolución de ejercicios (que no es necesaria si se aprobaron ambos parciales)

Debido a que la CAC decidio que no haya mas una parte de programacion en esta materia, ni hay persona designada a tal efecto, no habra proyecto de programacion.

REGULARIDAD
Para regularizar los/as estudiantes deberán aprobar aprobar los dos parciales o sus respectivos recuperatorios.

PROMOCIÓN
La materia no se promociona.