

Programas de Estudios Modalidad Escolarizada
INGENIERIA EN COMPUTACIÓN

NOMBRE DE LA ASIGNATURA: Programación Competitiva

CICLO, ÁREA O MÓDULO: Tercer Semestre

CLAVE: COM-11305

OBJETIVO(S) GENERAL(S) DE LA ASIGNATURA:

La materia integra los conocimientos formales y herramientas adquiridos en los cursos previos de algoritmos y programas, estructuras de datos, álgebra, desarrollo de sistemas con orientación a objetos y de desarrollo de aplicaciones con el objetivo de desarrollar las habilidades para reconocer los modelos susceptibles de ser aplicados en el desarrollo de los programas a ser aplicados en diversas problemáticas. Identificación rápida de los tipos de problemas. Análisis de complejidad del problema y de las posibles soluciones. Trabajo en equipo.

TEMAS Y SUBTEMAS:

- I. Inmersión a los concursos de programación. El objetivo de la programación competitiva. Tips para ser un programador competitivo. Elaboración de casos de prueba. Empaquetado de los casos en archivos. Uso de redirecciones y comparación automática de resultados.
- II. Los concursos de programación. Concursos y plataformas. Criterios y penalizaciones. Problemas Ad-hoc
- III. El lenguaje de programación C. Pre procesamiento, compilación y liga de objetos con bibliotecas. Diferencias entre funciones y macro sustituciones. Generación del ejecutable. Compilación, liga y ejecución de programas. Preprocesador. Tipos de datos, 32 y 64 bits. Arreglos y apuntadores. Uso del heap y del stack. Depuración de programas.
- IV. C++ y los objetos. C++ como pre procesado de C. Uso de STL. Sobrecarga de operadores y conveniencia de uso. Diferencias entre los lenguajes C, C++ y Java.
- V. STL y genéricos en java.util como bibliotecas de estructuras de datos. Estructuras de datos lineales. Estructuras de datos no lineales. Implementación de estructuras ad-hoc de datos: Grafos, Union-Find y conjuntos disjuntos, Segment Tree y Binary Indexed Tree (BIT).
- VI. Técnicas de resolución de problemas: Búsquedas. Aplicación del divide y vencerás. Uso de bisección y uso e implementación de búsqueda binaria. Algoritmos glotones. Programación dinámica. $O(n \log k)$ para la Subsecuencia incremental de valor máximo. (LIS). Problema de maximización de valor de artículos en la mochila. Subset Sum.
- VII. Implementación de algoritmos de grafos. Recorrido de Grafos. Depth First Search (DFS). Breadth First Search (BFS). Componentes conexas (Undirected Graphs). Flood Fill - Labeling/Coloring Connected Components. Ordenamiento topológico (Directed Acyclic Graph). Bipartite Graph Check. Graph Edges Property Check via DFS Spanning Tree. Finding Articulation Points and Bridges (in an Undirected Graph). Finding Strongly Connected Components (in a Directed Graph). Árbol de expansión mínimo (Minimum Spanning Tree). Kruskal's Algorithm. Prim's Algorithm. Caminos más cortos con un solo origen (Single-Source Shortest Paths). SSSP on Unweighted Graph. SSSP on Weighted Graph. SSSP on Graph with Negative Weight Cycle. Caminos más cortos entre todos los nodos (All-Pairs Shortest Paths). Floyd Warshall's DP Solution. Maximum Flow. Ford Fulkerson's Method. Edmonds Karp's. Grafos especiales. Directed Acyclic Graph. Trees. Eulerian Graph. Bipartite Graph.
- VIII. Aritmética entera. Clase Big Integer de Java. Long long en C. Combinatoria. Fibonacci Numbers. Binomial Coefficients. Catalan Numbers. Elementos de la Teoría de Números. Números primos. Greatest Common Divisor (GCD) & Least Common Multiple (LCM). Factorización de eficiente de números. Uso de factores primos. Funciones que utilizan factores primos. Aritmética modular. Extended Euclid: Solving Linear Diophantine Equation. Elementos de teoría de probabilidad. Cycle-Finding. Solution using Efficient Data Structure. Floyd's Cycle-Finding Algorithm. Teoría de Juegos. Árboles de decisión. Conceptos matemáticos para encontrar una solución de complejidad constante o lineal. Juego del Nim. Exponenciación de matrices. Exponenciación binaria (logarítmica). SquareMatrix Exponentiation.

- IX. Procesamiento de cadenas de texto. StringMatching. Knuth-Morris-Pratt (KMP) Algorithm. StringMatching in a 2D Grid. String Processing con programación dinámica. String Alignment (Edit Distance). Longest Common Subsequence. Palindrome. Suffix Trie/Tree/Array. Suffix Trie. Suffix Tree. Suffix Array.
- X. Geometría Computacional. Objetos geométricos básicos. Puntos. Líneas. 2D: Círculos.Triángulos. Cuadrilátero. 3D: Esferas. Otros. Polígonos. Representación. Perímetro de un polígono. Área de un polígono. Checking if a Polygon is Convex. Checking if a Point is Inside a Polygon. Cutting Polygon with a Straight Line. Finding the Convex Hull of a Set of Points.
- XI. Técnicas Avanzadas de solución de problemas. Descomposición de problemas. Dos problemas en uno: Búsqueda binaria + (Greedy ó función ó simulación). SSSP y DP. Varios algoritmos sobre grafos. Varias técnicas matemáticas. Tres problemas en uno: Factores primos, DP, Búsqueda binaria. Simulación completa, búsqueda binaria, Greedy. Técnicas avanzadas. Informed Search: A*. Depth Limited Search. Iterative Deepening Search. Iterative Deepening A* (IDA*). Técnicas avanzadas de programación dinámica. DP + bitmask. Chinese Postman/Route Inspection Problem. Compilation of Common DP States. Correcta representación de estados. Minimización de parámetros en los estados. Isomorfismo de grafos e identificación de algoritmos de solución.

ACTIVIDADES DE APRENDIZAJE:

Exposición de los temas en clase. Lecturas seleccionadas. Solución de problemas en clase. Solución de problemas fuera de clase. Análisis de las soluciones, memoria utilizada y tiempo de ejecución.

EVALUACIÓN DEL CURSO:

Durante el curso se realizarán dos exámenes parciales y uno final. Para obtener la calificación final se promediarán las calificaciones de todos los exámenes. Los exámenes se realizarán en la de los concursos para reforzar los temas y también se asignará una lista de ejercicios disponibles en la bibliografía principal. Los problemas serán revisados por medio de un juez automático en línea. Un porcentaje de la calificación se asignará en virtud del puntaje obtenido en los problemas resueltos.

BIBLIOGRAFÍA:

- **Competitive programming 2** : this increases the lower bound of programming contest : again / Steven Halim ; Felix Halim., [Singapore : National University of Singapore], 2011.
 - **Introduction to algorithms** / Thomas H. Cormen ... [et al.]. , 3rd ed. , Cambridge, Mass. : MIT Press, c2009.
 - **Programming pearls** / Jon Louis Bentley., 2nd ed. New York : ACM Press : Addison-Wesley, c2000.
 - **Computational geometry : algorithms and applications** / Mark de Berg ... [et al.], Algorithms and applications., , 2nd rev. ed., Berlin : Springer, c2000.
 - **The art of computer programming**, Knuth, Donald Ervin, 1938, 2nd. ed., Reading, Mass. : Addison-Wesley, 1973.
 - **Graphs, networks and algorithms** / Dieter Jungnickel ; tr. from the German by Tilla Schade., LinkJungnickel, Dieter, Berlin : Springer, 1999.
-