Análisis y diseño de algoritmos (3009430)

Objetivo del curso (oficial)

Cualquiera con una experiencia mínima en algoritmos y programación sabe que para un mismo problema pueden existir innumerables algoritmos de solución. Tales soluciones pueden variar en diversos aspectos, siendo de particular interés la eficiencia en términos de su costo computacional.

En el contexto de dicha eficiencia se puede hablar de algoritmos "mejores" que otros, siendo el objetivo principal de esta asignatura brindarle elementos al estudiante para analizar y diseñar algoritmos eficientes.

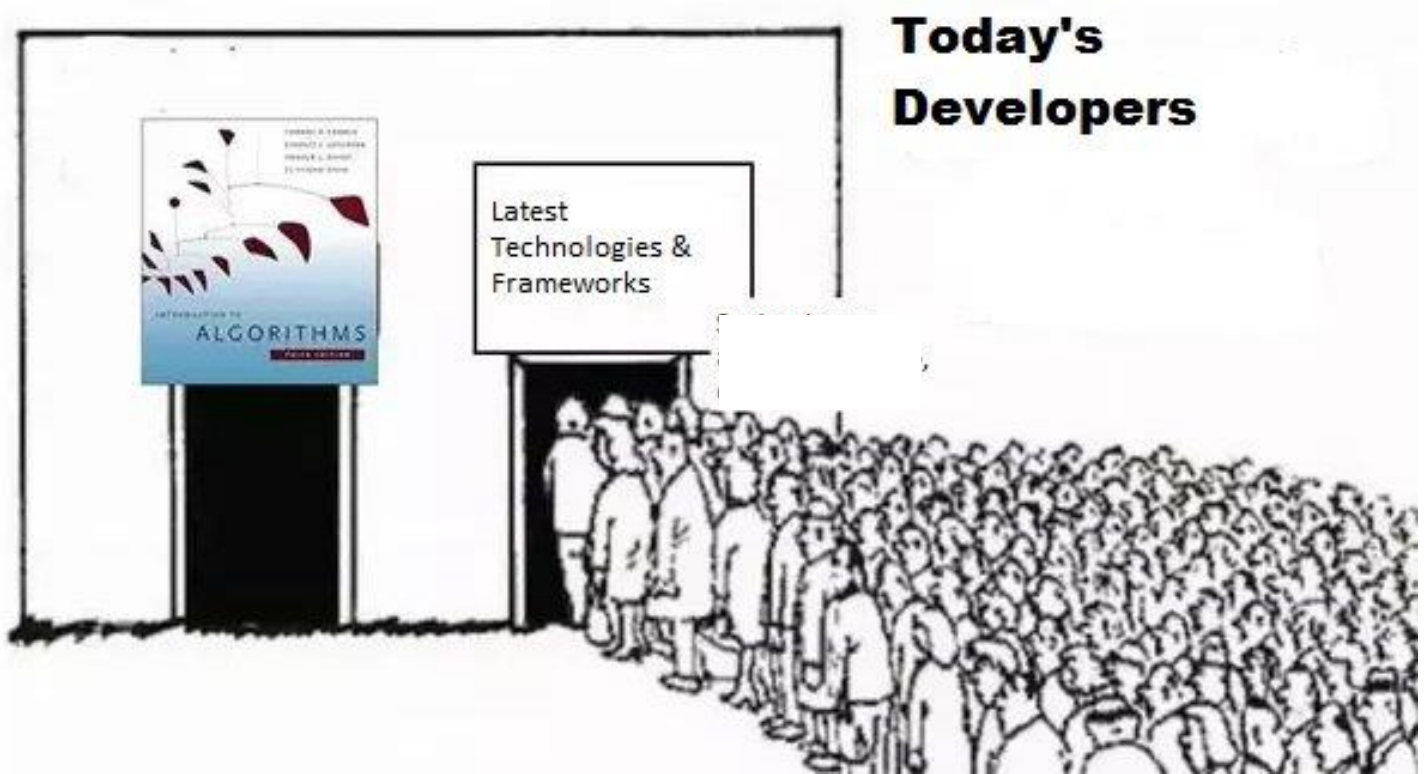
Objetivos del curso (extra-oficial)

- Ofertar una asignatura disciplinar optativa para Ingeniería de Sistemas e Informática. De igual manera que sirva en el componente electivo de otros programas de la Universidad Nacional de Colombia.
- Brindar un espacio para que los estudiantes a quienes les guste la programación mejoren sus habilidades
- Preparar a los estudiantes para participar en el circuito colombiano de maratones de programación y demás competencias, y llevar a la Sede Medellín a los primeros lugares
- Mejorar las probabilidades de vinculación laboral en empresas de tecnología importantes, o por qué no, de creación de nuevas empresas

¿Por qué estudiar algoritmos?

- ✓ Es importante para comprender conceptos y problemas de muchas otras áreas de computación y de ingeniería en general: redes y telecomunicaciones, criptografía, optimización, etc.
- ✓ Da “super poderes” en programación
- ✓ Es desafiante y satisfactorio

Ejemplo: ¿Cómo podemos contar un número muy grande de asistentes a un evento?



Pre-requisitos

- ✓ Excelentes bases de programación y estructuras de datos:
  - Recursividad
  - Arreglos y listas, Pilas y colas, Grafos, Árboles binarios, Montículos binarios
- ✓ Conocimientos mínimos de POO y buen manejo de al menos un lenguaje imperativo objetual (Python, Java, o C++)
- ✓ Conocimientos de cálculo: función logaritmo, series, integrales (básico)
- ✓ Conocimientos de estadística: valor esperado, variables aleatorias discretas (básico)

Metodología

Clases “magistrales”, no hay práctica, y este no es un curso de programación.

Evaluación:

- 4 Talleres prácticos, individuales y continuos con fechas de corte, de 20% cada uno

  Calificación de cada taller = $5.0 * \frac{ejercicios\ realizados}{ejercicios\ propuestos}$

  Cantidad de ejercicios por taller: 5
  Cantidad total de ejercicios en el semestre: 60

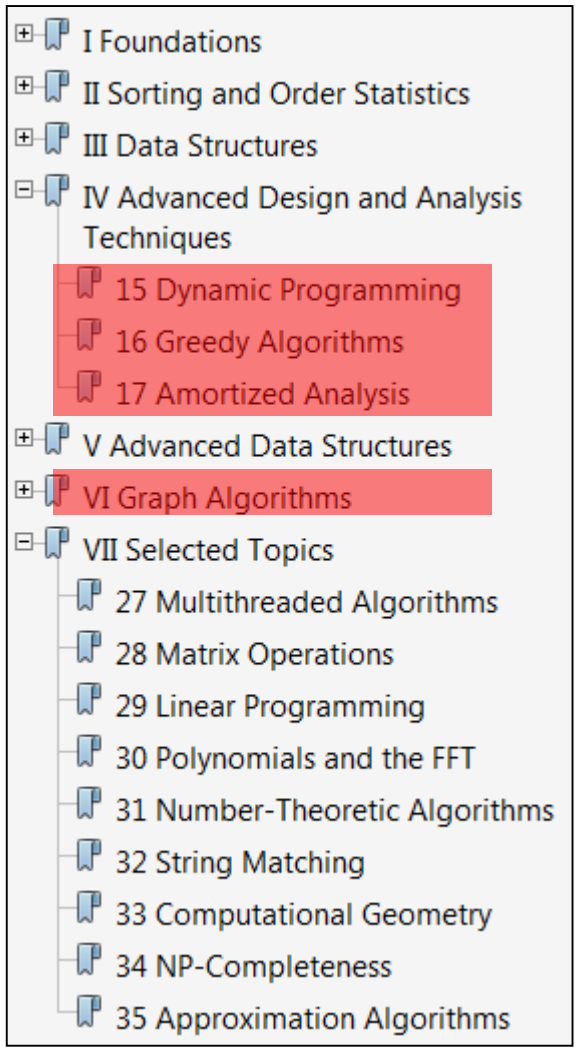
- 1 Trabajo final con exposición oral de 20%

Programa calendario

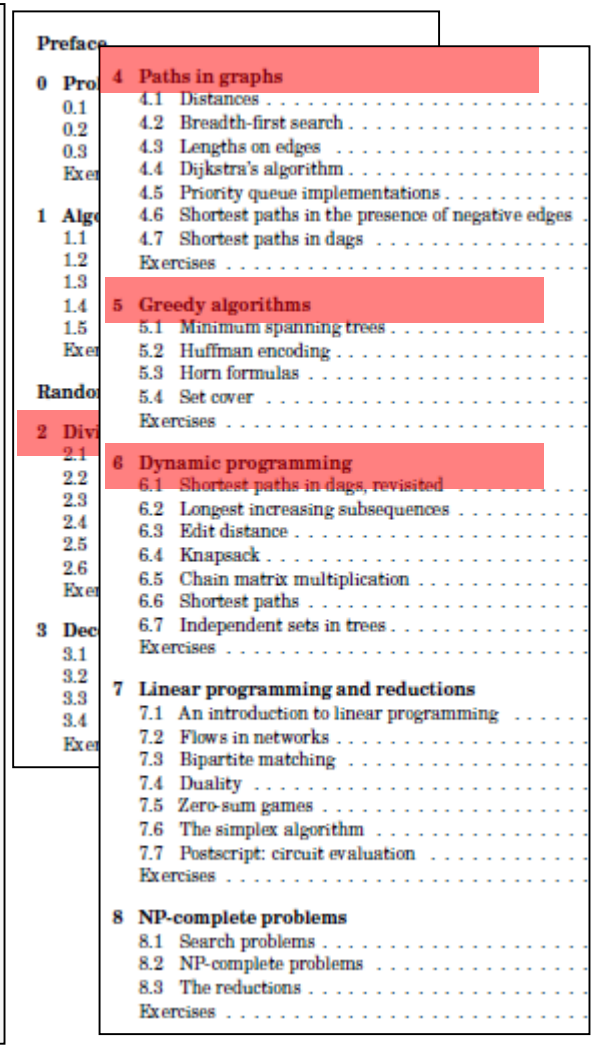
Sesión	Módulo	Contenido
1	Complejidad de algoritmos	Introducción Notación Big O Comparación de ordenes de complejidad
2	Algoritmos ad-hoc y clasificación de problemas	Cantidad de divisores Algoritmo de Euclides Criba de Eratóstenes Problemas P y NP
3[1]	Búsqueda exhaustiva	Cuadrados mágicos Ocho reinas Travel Salesman Problem Suma de subconjunto Coloreado de grafos
4	Algoritmos voraces	Cambio de monedas Planificación de tareas Programación de procesos unitarios Círculos de Malfatti
5		Selección de actividades Código de Huffman
6[2]		Algoritmo de Dijkstra Algoritmo de Prim Algoritmo de Edmonds–Karp
7	Programación dinámica	Máximo conjunto independiente Backtracking Corte de cable Knapsack
8		Cambio de monedas con programación dinámica Mayor subsecuencia común Alineación de secuencias Algoritmo de Floyd-Warshall
9[3]		Multiplicación en cadena de matrices Algoritmo de Prim
10	Divide y vencerás	MergeSort, Método maestro Búsqueda binaria
11		Cantidad de inversiones de un arreglo QuickSort Estadístico de orden k
12[4]		Potenciación recursiva Fibonacci matricial Multiplicación de Karatusuba Multiplicación de matrices de Strassen Pares más cercanos
13	Aplicaciones	
14	Trabajo final	

Temas “gruesos” en algoritmia

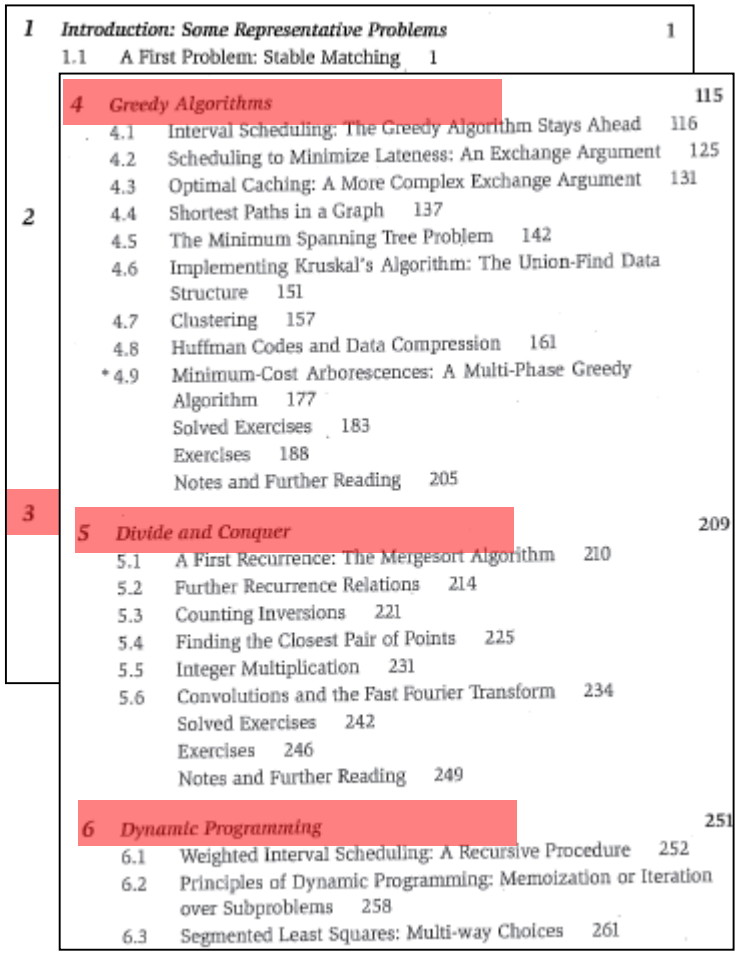
Introduction to algorithms, Cormen et al.



Algorithms, Dasgupta et al.



Algorithm design, Kleinberg and Tardos



Consideraciones importantes

- La motivación para tomar este curso debería ser una sola: aprender
- Relación de confianza profesor-estudiante y queda terminantemente prohibido copiar códigos de colegas en los ejercicios (se puede discutir y trabajar en grupos pero cada quien debe escribir su propio código)
- El profesor no se las sabe todas
- Casi nada del contenido del curso es original
- Las ayudas del curso son para aprovecharlas: grupo, monitor, asesorías, etc.
- Haremos uso de conceptos, teoremas, y métodos probados, más no nos vamos a concentrar necesariamente en su demostración
- Este curso es un punto de partida, el perfeccionamiento de las habilidades de programación depende de cada uno, de hecho, aún quedarían muchos temas por cubrir

Asesorías

Profesor

Julián Moreno, jmoreno1@unal.edu.co

Lunes 14:00 – 16:00, M8A-311

Monitores

David Antonio Aristizabal, daaristizabalg@unal.edu.co

Dia, hora y modalidad por definir

Grupo de WhatsApp

https://chat.whatsapp.com/KDoqRMIcixN6tNMTUU2xmf