

# ANÁLISIS DE ALGORITMOS

## NOTAS DE CLASE

DR. FELIPE CONTRERAS  
UACM, 2016-1

### 1. OBJETIVOS DEL CURSO ANÁLISIS DE ALGORITMOS

1. Complejidad intuitiva
  - a) Aprenderá el concepto de paso algorítmico
  - b) Aprenderá el concepto de unidad de memoria algorítmica
  - c) Conocerá el concepto de complejidad espacial y temporal a través de numerosos ejemplos
  - d) Distinguirá la complejidad de diferentes algoritmos para resolver un mismo problema
  - e) Aprenderá a medir la complejidad en el mejor y en el peor de los casos
  - f) Conocerá como calcular la complejidad de algoritmos en el caso promedio
2. Complejidad asintótica
  - a) Aprenderá el concepto de complejidad asintótica algebraicamente y las notaciones  $O$ ,  $\Omega$ ,  $o$ ,  $\omega$  y  $\Theta$
  - b) Aprenderá a calcular la complejidad asintótica de algoritmos
3. Clases de problemas
  - a) Conocerá el concepto de transformación de problemas y calculará complejidades con esta técnica
  - b) Conocerá los conjuntos de problemas  $P$ ,  $NP$  y  $NP$ -completos.
4. Diseño de algoritmos  
Conocerá y ejercitará el diseño de algoritmos con las siguientes técnicas
  - a) Exhaustivos
  - b) Divide y vencerás
  - c) Vuelta atrás (backtracking)
  - d) Recursividad y recursividad de cola
  - e) En árboles
  - f) Geométricos y/o en gráficas
  - g) Glotones (greedy)
  - h) Aleatorios (montecarlo)
  - i) Aproximados
  - j) Distribuidos

## 2. COMENZANDO

### 2.1. Algunas preguntas que debes contestarte al principio de este curso.

- ¿Para qué sirve el análisis de algoritmos?
- ¿Qué es una estructura de datos?, ¿Por qué es importante escoger la adecuada para un problema dado?
- ¿Qué se entiende por complejidad algorítmica? ¿Qué tipos de complejidad hay? ¿Cómo se mide?

### 2.2. Ejercicios y Tareas. Listado de ejercicios y tareas para el curso de Análisis de Algoritmos.

Descripción:

1. Los ejercicios NO tienen calificación pero se te recomienda hacerlos todos, ya que la próxima tarea o parcial probablemente salga de estos
2. Las tareas SI tienen calificación
3. Las tareas se entregan una semana después de que se solicitan
4. Las tareas se entregan por correo a la dirección [hobber.mallow@gmail.com](mailto:hobber.mallow@gmail.com)

**Tarea 1 :** 1. Escribir el programa de ordenamiento por inserción en lenguaje C o Java (NOTA: en adelante cuando se te pida programar, lo debes hacer en el lenguaje de tu preferencia (C, Java, Python, ...), siempre y cuando el programa NO se haya visto en clase en ese lenguaje. Por ejemplo, si en clase o las notas el programa se vió en Python, debes programarlo en cualquier otro lenguaje excepto Python).

- 2.
- 3.