

Algoritmos y Complejidad

Curso 2017

Pablo R. Fillottrani

Depto. Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Primer Cuatrimestre 2017



Resumen

Objetivos

Cronograma

Cursado

Recursos



- ▶ introducir y aplicar algunas técnicas de análisis de algoritmos
- ▶ ilustrar su uso a través de ejemplos prácticos en diversas áreas de las Ciencias de la Computación
- ▶ conocer los elementos básicos y problemas de la Complejidad Computacional

No se pretende memorizar algunas recetas, sino aprender a aplicar algunas técnicas.



1. Introducción (1 clase)
 - 1.1 algoritmos y algoritmia
 - 1.2 problemas e instancias
 - 1.3 tipos de análisis de eficiencia
 - 1.4 algunos ejemplos
2. Técnicas y Herramientas (2 clases)
 - 2.1 técnicas de demostración
 - 2.2 herramientas matemáticas básicas
 - 2.3 notación asintótica
 - 2.4 análisis de algoritmos por estructuras de control
 - 2.5 estructuras de datos, algoritmo *Heapsort*
 - 2.6 resolución de recurrencias



- 3. Algoritmos “Greedy” (1 clase)
 - 3.1 generalidades
 - 3.2 problema de la mochila
 - 3.3 scheduling de procesos
- 4. Algoritmos “Dividir y Conquistar” (3 clases)
 - 4.1 generalidades
 - 4.2 ordenamiento: mergesort y quicksort
 - 4.3 elemento mediano
 - 4.4 multiplicación de matrices: Strassen
 - 4.5 par de puntos más cercanos
 - 4.6 criptografía – exponenciación modular



5. Programación Dinámica (3 clases)

5.1 generalidades

5.2 problema del cambio

5.3 problema de la mochila

5.4 caminos más cortos

5.5 producto de cadenas de matrices (triangularización optimal de polígonos)

5.6 viajante

6. Algoritmos de Grafos (5 clases)

6.1 generalidades.

6.2 árboles de cubrimiento minimales: algoritmos de Kruskal y Prim

6.3 caminos más cortos con origen único: algoritmo de Dijkstra

6.4 problema del viajante

6.5 Recorridos. Propiedades.

6.6 orden topológico

6.7 componentes fuertemente conexos

6.8 puntos de articulación y puentes

6.9 flujo máximo



7. Análisis Amortizado (2 clases)

7.1 uso y principios. Formas de análisis.

7.2 tabla dinámica

7.3 skew heaps

7.4 heaps de Fibonacci

8. Algoritmos Probabilísticos (2 clases)

8.1 introducción

8.2 clasificación

8.3 análisis probabilístico

8.4 ejemplos



9. Complejidad Computacional (5 clases)

9.1 objetivos, conceptos básicos.

9.2 clases de complejidad.

9.3 clase **P**.

9.4 clase **NP**.

9.5 clase **NPC**.

9.6 problema $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$.

9.7 algoritmos de aproximación.

9.8 otras clases de complejidad



- ▶ dos parciales, cada uno con su recuperatorio. Calificación: A-B-C-D-E
- ▶ un proyecto obligatorio, con su re-entrega en caso de ser necesario. Calificación: aprobado, desaprobado
- ▶ fechas en la página web
- ▶ hay promoción para aquellos que cursan y no hayan obtenido dos C en sus parciales. Se tomará un tercer parcial sobre los temas no incluidos en los dos primeros parciales



► Bibliografía básica

- *Introduction to Algorithms, 3rd. ed*, T. Cormen, C. Leiserson, R. Rivest, C. Stein. MIT Press 2009.
- *P, NP, and NP-completeness*, O.Goldreich. Cambridge 2010.



► Bibliografía adicional

- *The Design and Analysis of Algorithms*, A. Levitin. Addison Wesley 2003.
- *Fundamentals of Algorithms*, G. Brassard, P. Bratley. Prentice Hall 1996.
- *Algorithmics, the Spirit of Computing, 2nd. ed*, D. Harel. Addison Wesley 1992.
- *Computational Complexity, A Modern Approach*, S. Arora, B. Barak. Cambridge 2009.
- *Introduction to the Theory of Complexity*, P. Bovet, P. Crescenzi. Prentice Hall 1993.
- *Computational Complexity*, C. Papadimitriou. Addison Wesley 1994.
- *The Art of Computer Programming, vol. I-III*, D. Knuth. Addison Wesley.
- *Analysis of Algorithms, 2nd ed*, R. Sedgewick. Addison Wesley 2013.



- ▶ cualquier otro libro avanzado sobre algoritmos es fuente de técnicas y ejemplos

- ▶ **Página web del curso**

cs.uns.edu.ar/~prf/teaching/AyC17/

- ▶ para transparencias, prácticos, enunciado del proyecto, links a sitios de interés, noticias, etc.

