

## INDICE

<b>Índice de algoritmos</b>	XVII
<b>Presentación</b>	XXI
<b>Capítulo 1. Introducción a la programación</b>	1
1.1. Introducción a los problemas	
1.1.1. Problemas, algoritmos y programas	3
1.1.2. Aspectos de la resolución de problemas	4
1.2. Introducción a los algoritmos y programas	
1.2.1. Sintaxis y semántica de un lenguaje	7
1.2.2. Paradigmas de programación	8
1.2.2. Fases en el desarrollo de un algoritmo	16
1.2.3. Corrección y eficiencia de algoritmos	17
1.2.4. Diseño de algoritmos	20
1.2.5. El estilo en la algorítmica	22
Notas bibliográficas	23
<b>Capítulo 2. Especificaciones de algoritmos funcionales</b>	25
2.1. Elementos básicos de especificación	
2.1.1. Identificadores	27
2.1.2. Subprogramas y funciones	29
2.1.3. Clases de especificaciones	33
2.2. Tipos de datos	
2.2.1. Tipos de datos predefinidos	35
2.2.2. Especificaciones de tipos de datos	41
2.3. Especificación con condiciones	53
2.3.1. Lógica matemática	54
2.3.2. Especificación funcional con condiciones	58
2.4. Especificación funcional operacional	62
2.4.1. Expresiones básicas	63
2.4.2. Recursividad	66
2.4.3. Ajuste de patrones	72
2.4.4. Definiciones locales	75
2.4.5. Funciones auxiliares	79
2.5. Normas de especificación	82
Notas bibliográficas	84
Ejercicios propuestos	85
<b>Capítulo 3. Especificaciones de algoritmos imperativos</b>	89
3.1. Elementos de especificación imperativa	
3.1.1. Procedimientos	90
3.1.2. Tipos de datos imperativos	91
3.1.3. Especificación imperativa con condiciones	98
3.2. Especificación imperativa operacional	
3.2.1. Declaraciones e instrucciones básicas	101
3.2.2. Iteración	106

3.2.3. Llamada a procedimiento	114
3.2.4. Recursividad	118
3.3. Elementos de estilo de especificación operacional imperativa	128
Notas bibliográficas	129
Ejercicios propuestos	130
<b>Capítulo 4. Estructuras de datos</b>	133
4.1. Estructuras y tipos de datos	135
4.2. Listas	137
4.3. Árboles	141
4.4. Grafos	145
4.5. Otras estructuras	
4.5.1. Pilas	151
4.5.2. Colas	152
4.5.3. Tablas	154
Notas bibliográficas	
Ejercicios propuestos	157
<b>Capítulo 5. Semántica y corrección de algoritmos</b>	161
5.1. Conceptos generales	
5.1.1. Semántica	163
5.1.2. Corrección	169
5.2. Semántica operacional funcional	172
5.2.1. Reescritura	173
5.2.2. Evaluación de expresiones	178
5.3. Semántica axiomática imperativa	182
5.4. Verificación de algoritmos	187
5.4.1. Principios de inducción	188
5.4.2. Verificación de algoritmos funcionales	191
5.4.3. Verificación de algoritmos imperativos	196
Notas bibliográficas	202
Ejercicios propuestos	203
<b>Capítulo 6. Complejidad de algoritmos</b>	207
6.1. Conceptos de complejidad algorítmica	
6.1.1. Evaluación de la eficiencia	209
6.1.2. Notaciones para expresar la complejidad tiempo: $O()$ , $U()$ y $\theta()$	211
6.1.3. Propiedades de la notación $O()$	213
6.2. Análisis de complejidad de algoritmos no recursivos	
6.2.1. Cálculo de la complejidad de las instrucciones en algoritmos imperativos	215
6.2.2. Cálculo de la complejidad de las expresiones en algoritmos funcionales	223
6.3. Análisis de complejidad de algoritmos recursivos	
6.3.1. Métodos para el cálculo de la complejidad de algoritmos recursivos	225
6.3.2. Series recurrentes de utilidad	235
Notas bibliográficas	
Ejercicios propuestos	236

<b>Capítulo 7. Algoritmos sencillos</b>	239
7.1. Algoritmos sobre enteros	
7.1.1. Cambio de base de un entero	241
7.1.2. Máximo común divisor de dos enteros	244
7.1.3. Generación de números primos	248
7.2. Algoritmos sobre vectores	
7.2.1. Búsqueda de un elemento en un vector	253
7.2.2. Ordenación de un vector	261
7.3. Algoritmos sobre arboles	
7.3.1. Recorrido de un árbol	272
7.3.2. Borrado de un elemento de un árbol de búsqueda	281
Notas bibliográficas	284
Ejercicios propuestos	285
<b>Capítulo 8. Técnicas de diseño de algoritmos</b>	291
8.1. Consideraciones generales	293
8.2. Método de fuerza bruta	295
8.3. Divide y vencerás	
8.3.1. Esquema de divide y vencerás	298
8.3.2. Elaboración de un calendario deportivo	300
8.3.3. Ordenación de un vector por mezcla	304
8.3.4. Transposición de un matriz cuadrada	306
8.3.5. Multiplicación de matrices	310
8.4. Técnicas de los parámetros acumuladores y de tabulación	
8.4.1. Parámetros acumuladores	314
8.4.2. Tabulación	317
8.5. Algoritmos voraces	
8.5.1. Esquema voraz	321
8.5.2. Desglose en monedas	324
8.5.3. Problema de la mochila	326
8.5.4. Árbol recubridor de coste mínimo	330
8.5.5. Caminos más cortos desde un solo origen	334
8.6. Programación dinámica	
8.6.1. Principios de Programación dinámica	338
8.6.2. Problema de la mochila 0-1	343
8.6.3. Camino de coste mínimo en un grafo multietapa	345
8.7. Búsqueda con retroceso	
8.7.1. Problemas de búsqueda	350
8.7.2. Esquema de búsqueda con retroceso	356
8.7.3. Solución a un laberinto	359
8.7.4. Problema de las n reinas	362
8.7.5. Problema de la mochila 0-1	365
8.8. Transformaciones del dominio	370
Notas bibliográficas	375
Ejercicios propuestos	376
<b>Capítulo 9. Algoritmos elaborados</b>	383

9.1. Algoritmos sobre estructuras secuenciales (vectores, listas y tablas)	
9.1.1. Partición de un vector	384
9.1.2. Subsecuencia monótona de mayor longitud	386
9.1.3. Ordenación por incrementos decrecientes	389
9.1.4. Ordenación rápida de una lista	392
9.1.5. Búsqueda de un elemento en una tabla hash	394
9.2. Algoritmos sobre arboles	
9.2.1. Creación de un árbol óptimo de búsqueda	396
9.2.2. Inserción de un elemento en un árbol AVL	402
9.2.3. Borrado genérico de un elemento en un árbol binario	407
9.3. Algoritmos sobre grafos	
9.3.1. Caminos más cortos entre los nodos de un grafo	409
9.3.2. Cierre transitivo de un grafo	411
9.3.3. Número de caminos mínimos de un grafo	412
9.4. Otros algoritmos	
9.4.1. Pertenencia de un punto a un polígono	414
9.4.2. Generación de números aleatorios	418
9.4.3. Cálculo de áreas por estimación	424
9.4.4. Unificación de términos	426
Notas bibliográficas	429
Ejercicios propuestos	430
<b>Capítulo 10. Una introducción a los problemas NP completos</b>	433
10.1. Ineficiencia e intratabilidad	
10.1.1. Tiempos de ejecución no razonables	434
10.1.2. Problemas P y NP	435
10.1.3. Reducciones en tiempo polinómico y NP completitud	437
10.1.4. Teorema de Cook	438
10.2. Demostración de la NP completitud de problemas	
10.2.1. Ejemplos de problemas NP completos	441
10.2.2. Algunas técnicas para demostrar la NP completitud de problemas	446
10.3. Técnicas para el tratamiento de problemas NP completos	
10.3.1. Búsqueda con retroceso. Bifurcación y acotación	449
10.3.2. Algoritmos de aproximación	453
10.4. Más allá de la NP completitud	455
Notas bibliográficas	457
Ejercicios propuestos	458
<b>Apéndices</b>	
<b>A. Sintaxis del lenguaje de especificación</b>	459
<b>B. Ejemplos de algoritmos en SML y modula-2</b>	473
<b>C. Tabla de caracteres ASCH</b>	485
<b>Bibliografía</b>	487
<b>Índice alfabético</b>	497