

| **Chapter 2: Arrays, Iteration, Invariants** | **Chapter 3: Lists, Recursion, Stacks, Queues** | **Chapter 4: Searching** |

| ----- | ----- | ----- |

| 2 Arrays | 3.1 Linked Lists | 4 Searching |

| 2.1 Arrays | 3.2 Recursion | 4.1 Requirements... |

| 2.2 Loops and Iteration | 3.3 Stacks | 4.2 Specification... |

| 2.3 Invariants | 3.4 Queues | 4.3 Linear Search |

| | 3.5 Doubly Linked Lists | 4.4 Binary Search |

| | 3.6 Advantages of Abstract Data Types | |

| | | |

| **Chapter 5: Efficiency and Complexity** | **Chapter 6: Trees** | **Chapter 7: Binary Search Trees** |

| 5 Efficiency and Complexity | 6.1 General Specification of Trees | 7.1 Searching with... |

| 5.1 Time versus space complexity | 6.2 Quad-trees | 7.2 Search keys |

| 5.2 Worst versus average complexity | 6.3 Binary trees | 7.3 Binary search trees |

| 5.3 Concrete measures for performance | 6.4 Primitive operations on binary trees | 7.4 Building binary... |

| 5.4 Big-O notation for complexity class | 6.5 The height of a binary tree | 7.5 Searching a binary... |

| 5.5 Formal definition of complexity class | 6.6 The size of a binary tree | 7.6 Time complexity... |

| | 6.7 Implementation of trees | 7.7 Deleting nodes... |

| | 6.8 Recursive algorithms | 7.8 Checking whether... |

| | | 7.9 Sorting using... |

| | | 7.10 Balancing binary... |

| | | 7.11 Self-balancing AVL... |

| | | 7.12 B-trees |

| | | |

| **Chapter 8: Priority Queues and Heap Trees** | **Chapter 9: Sorting** | **Chapter 10: Hash Tables** |

| 8 Priority Queues and Heap Trees | 9 Sorting | 10 Hash Tables |

| 8.1 Trees stored in arrays | 9.1 The problem of sorting | 10.1 Storing data |

| 8.2 Priority queues and binary heap trees | 9.2 Common sorting strategies | 10.2 Table ADT |

| 8.3 Basic operations on binary heap trees | 9.3 How many comparisons must it take? | 10.3 Implementations... |

| 8.4 Inserting a new heap tree node | 9.4 Bubble Sort | 10.4 Hash Tables |

| 8.5 Deleting a heap tree node | 9.5 Insertion Sort | 10.5 Collision likelihood... |

| 8.6 Building a new heap tree from scratch | 9.6 Selection Sort | 10.6 A simple Hash... |

8.7 Merging binary heap trees	9.7 Comparison of $O(n^2)$ sorting algorithms	10.7 Strategies for...	
8.8 Binomial heaps	9.8 Sorting algorithm stability	10.8 Linear Probing...	
8.9 Fibonacci heaps	9.9 Treesort	10.9 Double Hashing...	
8.10 Comparison of heap time complexities	9.10 Heapsort	10.10 Choosing good...	
	9.11 Divide and conquer algorithms	10.11 Complexity of...	
	9.12 Quicksort		
	9.13 Mergesort		
	9.14 Summary of comparison-based sorting...		
	9.15 Non-comparison-based sorts		
	9.16 Bin, Bucket, Radix Sorts		
Chapter 11: Graphs	Chapter 12: Epilogue	Appendix A: Some Useful	
Formulae			
11 Graphs			