

1. Record Nr.	UNINA9911006689903321
Titolo	Introduction to Algorithms : A Comprehensive Guide for Beginners : Unlocking Computational Thinking
Pubbl/distr/stampa	Birmingham, : Packt Publishing, Limited, 2024
ISBN	9781836203872
Descrizione fisica	1 online resource (294 p.)
Disciplina	005.1
Soggetti	Computer algorithms Computer programming
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record. 6.3 Insertion Sort
Nota di contenuto	Intro -- Code Blocks Resource -- Premium Customer Support -- Who we are -- Our Philosophy -- Our Expertise -- Introduction -- Preface -- Introduction to the Book -- Who Should Read This Book -- How to Use This Book -- Chapter 1: Introduction to Algorithms -- 1.1 What is an Algorithm? -- 1.1.1 Characteristics of a Good Algorithm -- 1.1.2 How Algorithms are Used -- 1.1.3 Brief Summary and Some Points for Further Reflection -- 1.2 Importance of Algorithms in Computer Science -- 1.2.1 Algorithms Power Our Digital World -- 1.2.2 Algorithms Drive Efficiency -- 1.2.3 Algorithms Form the Basis of Advanced Fields -- 1.2.4 The Future Implications and Advancements in Algorithms -- 1.3 Fundamentals of Computational Thinking -- 1.3.1 Decomposition -- 1.3.2 Pattern Recognition -- 1.3.3 Abstraction -- 1.3.4 Algorithmic Thinking -- 1.3.5 Debugging and Iteration -- 1.4 Practice Problems -- Problem 1: Password Generator -- Problem 2: Calendar Events -- Problem 3: Building a Pyramid -- Problem 4: Text Compression -- 1.5 Chapter Summary -- Chapter 2: Pseudocode and Flowcharts -- 2.1 Understanding Pseudocode -- 2.1.1 Flexibility of Pseudocode -- 2.2 Understanding Flowcharts -- 2.3 Translating Real-World Problems into Pseudocode -- 2.4 Practice Problems -- Problem 1 -- Problem 2 -- Problem 3 -- Chapter 2 Summary -- Chapter 3: Algorithm Efficiency -- 3.1 Understanding Time Complexity -- 3.1.1 Big O notation -- 3.1.2

Difference between the best-case, average-case, and worst-case time complexity -- 3.2 Understanding Space Complexity -- 3.2.1  
 Caching/Memoization -- 3.3 Introduction to Big O Notation -- 3.3.1  
 What is Big O Notation? -- 3.3.2 Common Types of Time Complexities -- 3.3.3 Asymptotic Analysis -- 3.4 Practice Problems  
 Linear Search: -- Sum of Elements: -- Find Duplicate: -- Bubble Sort: -- Chapter 3 Summary -- Chapter 4: Basic Algorithm Types -- 4.1  
 Divide and Conquer Algorithms -- 4.2 Greedy Algorithms -- 4.2.1  
 What is a Greedy Algorithm? -- 4.2.2 Coin Change Problem -- 4.3  
 Dynamic Programming Algorithms -- 4.4 Recursive Algorithms -- 4.4.1 Tail Recursion -- 4.5 Practice Problems -- Problem 1: Binary Search (Divide and Conquer) -- Problem 2: Coin Change (Greedy Algorithm) -- Problem 3: Fibonacci Series (Dynamic Programming) -- Problem 4: Sum of Natural Numbers (Recursive Algorithm) -- Problem 5: QuickSort (Divide and Conquer) -- Problem 6: Implementing a Stack using Recursion (Recursive Algorithm) -- Chapter 4 Summary -- Chapter 5: Search Algorithms -- 5.1 Linear Search -- 5.1.1  
 Limitations of Linear Search -- 5.2: Binary Search -- 5.3 Hashing and Hash Tables -- 5.3.1 Collisions -- 5.4 Practice Problems -- Problem 1: Linear Search -- Problem 2: Binary Search -- Problem 3: Hashing -- Problem 4: Binary Search vs Linear Search -- Chapter 5 Summary -- Chapter 6: Sort Algorithms -- 6.1 Bubble Sort -- 6.1.1 When and Why to Use or Not to Use Bubble Sort -- 6.2 Selection Sort

---

## Sommario/riassunto

Discover the fundamentals and advanced concepts of algorithms with this comprehensive course. Learn about efficiency, types, design techniques, and real-world applications, and enhance your algorithmic knowledge. Key Features Basics to advanced algorithm design and applications, along with real-world applications Engaging exercises & case studies from the latest industry trends & practices for reinforcement Clear, step-by-step instructions for complex and advanced topics Book Description Begin your journey into the fascinating world of algorithms with this comprehensive course. Starting with an introduction to the basics, you will learn about pseudocode and flowcharts, the fundamental tools for representing algorithms. As you progress, you'll delve into the efficiency of algorithms, understanding how to evaluate and optimize them for better performance. The course will also cover various basic algorithm types, providing a solid foundation for further exploration. You will explore specific categories of algorithms, including search and sort algorithms, which are crucial for managing and retrieving data efficiently. You will also learn about graph algorithms, which are essential for solving problems related to networks and relationships. Additionally, the course will introduce you to the data structures commonly used in algorithms. Towards the end, the focus shifts to algorithm design techniques and their real-world applications. You will discover various strategies for creating efficient and effective algorithms and see how these techniques are applied in real-world scenarios. By the end of the course, you will have a thorough understanding of algorithmic principles and be equipped with the skills to apply them in your technical career. What you will learn Understand the basics of algorithms and their significance Evaluate the efficiency of different algorithms Apply various types of algorithms to solve complex problems Utilize graph algorithms for network-related issues Implement appropriate data structures for algorithm optimization Design efficient algorithms for real-world applications Who this book is for This course is designed for a wide range of learners, including technical professionals looking to enhance their algorithmic knowledge, computer science students seeking a deeper understanding of

algorithm principles, and software developers aiming to improve their coding efficiency. Additionally, it is suitable for data scientists and analysts who need to apply algorithms to data management and analysis tasks, educators looking for comprehensive teaching material on algorithms, and hobbyists interested in expanding their technical skill set.

---