

Introduction to Algorithms Notes

Jean philo Gong

September 19, 2023

Contents

1	The Role of Algorithms in Computing Notes	7
1.1	Algorithms	7
1.1.1	What kinds of problems are solved by algorithms	7
1.1.2	Data structures	7
1.1.3	Technique	7
1.2	Algorithms as a technology	7
2	Getting Started	9
2.1	Insertion sort	9
2.1.1	Insertion Sort Algorithm	9
2.1.2	Code Implementation	9
2.2	Analyzing algorithms	10
2.3	Designing algorithms	10
3	Growth of Functions	11
3.1	Asymptotic notation	11
3.2	Standard notations and common functions	11
4	Divide-and-Conquer	13
4.1	The maximum-subarray problem	13
4.2	Strassen's algorithm for matrix multiplication	13
4.3	The substitution method for solving recurrences	13
4.4	The recursion-tree method for solving recurrences	13
4.5	The master method for solving recurrences	13
4.6	Proof of the master theorem	13
5	Probabilistic Analysis and Randomized Algorithms	15
6	Heapsort	17
7	Quicksort	19
8	Sorting in Linear Time	21
9	Medians and Order Statistics	23

10 Elementary Data Structures	25
11 Hash Tables	27
12 Binary Search Trees	29
13 Red-Black Trees	31
14 Augmenting Data Structures	33
15 Dynamic Programming	35
16 Greedy Algorithms	37
17 Amortized Analysis	39
18 B-Trees	41
19 Fibonacci Heaps	43
20 Van Emde Boas Trees	45
21 Data Structures for Disjoint Sets	47
22 Graph Algorithms	49
23 Minimum Spanning Trees	51
24 Single-Source Shortest Paths	53
25 All-Pairs Shortest Paths	55
26 Maximum Flow	57
27 Multithreaded Algorithms	59
28 Matrix Operations	61
29 Linear Programming	63
30 Polynomials and the FFT	65
31 Number-Theoretic Algorithms	67
32 String Matching	69
33 Computational Geometry	71
34 NP-Completeness	73

<i>CONTENTS</i>	5
35 Approximation Algorithms	75
36 Mathematical Background	77
37 Problems, Hints, and Solutions	79

Chapter 1

The Role of Algorithms in Computing Notes

1.1 Algorithms

An example of an algorithm is as follows:

Input: A sequence of n numbers (a_1, a_2, \dots, a_n) .

Output: A permutation (reordering) $(a'_1, a'_2, \dots, a'_n)$

such that $a'_1 \leq a'_2 \leq \dots \leq a'_n$.

1.1.1 What kinds of problems are solved by algorithms

1.1.2 Data structures

Definition:

A data structure is a way to store and organize data in order to facilitate access and modifications.

1.1.3 Technique

1.2 Algorithms as a technology

Chapter 2

Getting Started

2.1 Insertion sort

Input: A sequence of n numbers (a_1, a_2, \dots, a_n) .

Output: A permutation (reordering) $(a'_1, a'_2, \dots, a'_n)$
such that $a'_1 \leq a'_2 \leq \dots \leq a'_n$.

Methods

2.1.1 Insertion Sort Algorithm

The insertion sort algorithm can be broken down into the following steps:

1. Define a function to perform the insertion sort operation.
2. Loop starts from the second element.
3. Store the current element as the key.
4. Initialize j as the element just before i .
5. Move elements that are greater than the key to one position ahead of their current position.
6. Place the key in its correct position.

2.1.2 Code Implementation

The Python code for the insertion sort algorithm is given below:

```
def insertion_sort(arr):  
    for i in range(1, len(arr)):  
        key = arr[i]
```

```
j = i - 1
while j >= 0 and key < arr[j]:
    arr[j + 1] = arr[j]
    j -= 1
arr[j + 1] = key
```

2.2 Analyzing algorithms

2.3 Designing algorithms

Chapter 3

Growth of Functions

3.1 Asymptotic notation

3.2 Standard notations and common functions

Chapter 4

Divide-and-Conquer

- 4.1 The maximum-subarray problem
- 4.2 Strassen's algorithm for matrix multiplication
- 4.3 The substitution method for solving recurrences
- 4.4 The recursion-tree method for solving recurrences
- 4.5 The master method for solving recurrences
- 4.6 Proof of the master theorem

Chapter 5

Probabilistic Analysis and Randomized Algorithms

Chapter 6

Heapsort

Chapter 7

Quicksort

Chapter 8

Sorting in Linear Time

Chapter 9

Medians and Order Statistics

Chapter 10

Elementary Data Structures

Chapter 11

Hash Tables

Chapter 12

Binary Search Trees

Chapter 13

Red-Black Trees

Chapter 14

Augmenting Data Structures

Chapter 15

Dynamic Programming

Chapter 16

Greedy Algorithms

Chapter 17

Amortized Analysis

Chapter 18

B-Trees

Chapter 19

Fibonacci Heaps

Chapter 20

Van Emde Boas Trees

Chapter 21

Data Structures for Disjoint Sets

Chapter 22

Graph Algorithms

Chapter 23

Minimum Spanning Trees

Chapter 24

Single-Source Shortest Paths

Chapter 25

All-Pairs Shortest Paths

Chapter 26

Maximum Flow

Chapter 27

Multithreaded Algorithms

Chapter 28

Matrix Operations

Chapter 29

Linear Programming

Chapter 30

Polynomials and the FFT

Chapter 31

Number-Theoretic Algorithms

Chapter 32

String Matching

Chapter 33

Computational Geometry

Chapter 34

NP-Completeness

Chapter 35

Approximation Algorithms

Chapter 36

Mathematical Background

Chapter 37

Problems, Hints, and Solutions