



# TypeScript

## ALGORITHMS

Version 0.0.1



# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Algorithms and Data Structures</b>	<b>3</b>
2.1	Algorithm Analysis . . . . .	4
2.2	Bits . . . . .	4
2.2.1	Bit Parity . . . . .	4
2.2.2	Bit Shift Operator . . . . .	5
2.3	Recursion . . . . .	5
2.3.1	Fibonacci Sequence . . . . .	5
<b>3</b>	<b>Domain Specific</b>	<b>6</b>
3.1	Language . . . . .	6
3.1.1	This . . . . .	6
3.1.2	Event Loop . . . . .	6
3.1.3	Asynchronous Programming . . . . .	6
3.1.4	Runtime Environments . . . . .	6
<b>4</b>	<b>Appendix</b>	<b>7</b>
4.1	Resources . . . . .	7
4.2	Contributing . . . . .	7
4.2.1	Getting Started . . . . .	7

# Chapter 1

## Introduction

In-progress book about algorithms and data structures in TypeScript.



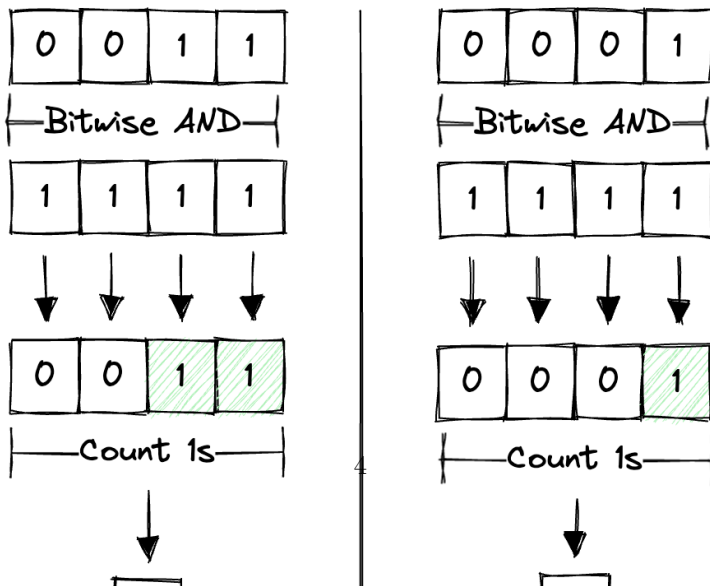
## Chapter 2

# Algorithms and Data Structures

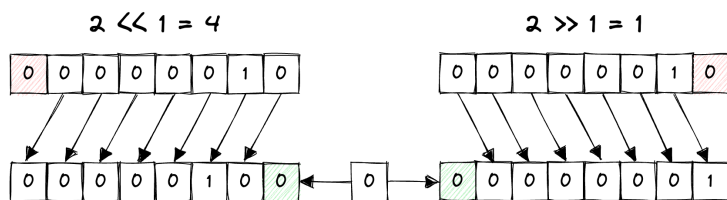
### 2.1 Algorithm Analysis

### 2.2 Bits

#### 2.2.1 Bit Parity

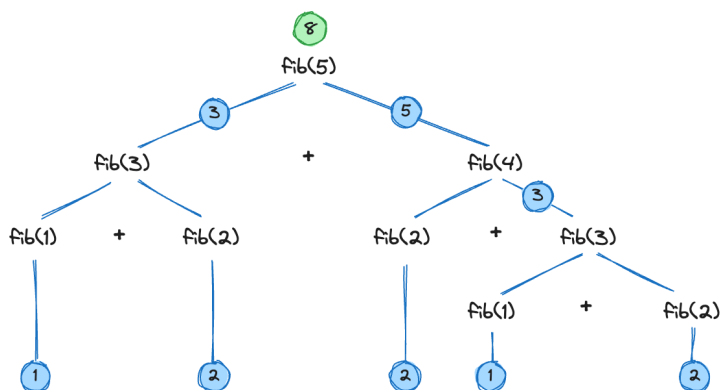


## 2.2.2 Bit Shift Operator



## 2.3 Recursion

### 2.3.1 Fibonacci Sequence



$$fib_n = fib_{n-2} + fib_{n-1}$$

# Chapter 3

## Domain Specific

### 3.1 Language

#### 3.1.1 This

#### 3.1.2 Event Loop

#### 3.1.3 Asynchronous Programming

##### 3.1.3.1 Promises

##### 3.1.3.2 Async/Await

#### 3.1.4 Runtime Environments

##### 3.1.4.1 Browser

##### 3.1.4.2 Server



# Chapter 4

# Appendix

## 4.1 Resources

- LeetCode
- Project Euler
- The Algorithm Design Manual
- Elements of Programming Interviews

## 4.2 Contributing

### 4.2.1 Getting Started

#### 4.2.1.1 Run Problem Tests

- Install Node Version Manager
- `yarn setup`
- `yarn test --all`

#### 4.2.1.2 Compile Book

- Install Pandoc (Homebrew)
- Install BasicTeX (Homebrew)

- Install fswatch (Homebrew)
- `make book`
- `open book/output/index.pdf`

#### 4.2.1.3 Editing Diagrams

- Install Excalidraw VSCode Extension

#### 4.2.1.4 Commands

Command	Description
<code>yarn setup</code>	Setup local development environment
<code>yarn test</code>	Run tests
<code>yarn gen:leetcode</code>	Generate new LeetCode problem
<code>yarn gen:project-euler</code>	Generate new Project Euler problem
<code>make book</code>	Compile book to various formats
<code>make watch</code>	Recompile book automatically when source files change