



# Algorithm

# Table of Contents

---

1. [Preface](#)
2. [Part I - Basics](#)
3. [Basics Data Structure](#)
  - i. [Linked List](#)
  - ii. [Binary Tree](#)
  - iii. [Binary Search Tree](#)
  - iv. [Huffman Compression](#)
  - v. [Priority Queue](#)
4. [Basics Sorting](#)
  - i. [Bubble Sort](#)
  - ii. [Selection Sort](#)
  - iii. [Insertion Sort](#)
  - iv. [Merge Sort](#)
  - v. [Quick Sort](#)
  - vi. [Heap Sort](#)
  - vii. [Bucket Sort](#)
  - viii. [Counting Sort](#)
  - ix. [Radix Sort](#)
5. [Basics Misc](#)
  - i. [Bit Manipulation](#)
  - ii. [Knapsack](#)
6. [Part II - Coding](#)
7. [String](#)
  - i. [strStr](#)
  - ii. [Two Strings Are Anagrams](#)
  - iii. [Compare Strings](#)
  - iv. [Anagrams](#)
  - v. [Longest Common Substring](#)
  - vi. [Rotate String](#)
  - vii. [Reverse Words in a String](#)
  - viii. [Valid Palindrome](#)
  - ix. [Longest Palindromic Substring](#)
8. [Integer Array](#)
  - i. [Remove Element](#)
  - ii. [Zero Sum Subarray](#)
  - iii. [Subarray Sum K](#)
  - iv. [Subarray Sum Closest](#)
  - v. [Recover Rotated Sorted Array](#)
  - vi. [Product of Array Exclude Itself](#)
  - vii. [Partition Array](#)
  - viii. [First Missing Positive](#)
  - ix. [2 Sum](#)
  - x. [3 Sum](#)
  - xi. [3 Sum Closest](#)
  - xii. [Remove Duplicates from Sorted Array](#)
  - xiii. [Remove Duplicates from Sorted Array II](#)
  - xiv. [Merge Sorted Array](#)
  - xv. [Merge Sorted Array II](#)
  - xvi. [Median](#)

9. Binary Search
  - i. Binary Search
  - ii. Search Insert Position
  - iii. Search for a Range
  - iv. First Bad Version
  - v. Search a 2D Matrix
  - vi. Find Peak Element
  - vii. Search in Rotated Sorted Array
  - viii. Find Minimum in Rotated Sorted Array
  - ix. Search a 2D Matrix II
  - x. Median of two Sorted Arrays
  - xi. Sqrt x
  - xii. Wood Cut
10. Math and Bit Manipulation
  - i. Single Number
  - ii. Single Number II
  - iii. Single Number III
  - iv. O1 Check Power of 2
  - v. Convert Integer A to Integer B
  - vi. Factorial Trailing Zeroes
  - vii. Unique Binary Search Trees
  - viii. Update Bits
  - ix. Fast Power
11. Linked List
  - i. Remove Duplicates from Sorted List
  - ii. Remove Duplicates from Sorted List II
  - iii. Remove Duplicates from Unsorted List
  - iv. Partition List
  - v. Two Lists Sum
  - vi. Two Lists Sum Advanced
  - vii. Remove Nth Node From End of List
  - viii. Linked List Cycle
  - ix. Linked List Cycle II
  - x. Reverse Linked List
  - xi. Reverse Linked List II
  - xii. Merge Two Sorted Lists
  - xiii. Merge k Sorted Lists
  - xiv. Reorder List
  - xv. Copy List with Random Pointer
  - xvi. Sort List
  - xvii. Insertion Sort List
  - xviii. Check if a singly linked list is palindrome
12. Binary Tree
  - i. Binary Tree Preorder Traversal
  - ii. Binary Tree Inorder Traversal
  - iii. Binary Tree Postorder Traversal
  - iv. Binary Tree Level Order Traversal
  - v. Maximum Depth of Binary Tree
  - vi. Balanced Binary Tree
  - vii. Binary Tree Maximum Path Sum
  - viii. Lowest Common Ancestor
13. Binary Search Tree
  - i. Insert Node in a Binary Search Tree

- ii. Validate Binary Search Tree
- iii. Search Range in Binary Search Tree
- iv. Convert Sorted Array to Binary Search Tree
- v. Convert Sorted List to Binary Search Tree
- vi. Binary Search Tree Iterator
- 14. Exhaustive Search
  - i. Subsets
  - ii. Unique Subsets
  - iii. Permutation
  - iv. Unique Permutations
  - v. Next Permutation
  - vi. Previous Permutation
  - vii. Unique Binary Search Trees II
  - viii. Permutation Index
  - ix. Permutation Index II
  - x. Permutation Sequence
  - xi. Palindrome Partitioning
- 15. Dynamic Programming
  - i. Triangle
  - ii. Backpack
  - iii. Minimum Path Sum
  - iv. Unique Paths
  - v. Unique Paths II
  - vi. Climbing Stairs
  - vii. Jump Game
  - viii. Word Break
  - ix. Longest Increasing Subsequence
  - x. Palindrome Partitioning II
  - xi. Longest Common Subsequence
  - xii. Edit Distance
  - xiii. Jump Game II
  - xiv. Best Time to Buy and Sell Stock
  - xv. Best Time to Buy and Sell Stock II
  - xvi. Best Time to Buy and Sell Stock III
  - xvii. Best Time to Buy and Sell Stock IV
  - xviii. Distinct Subsequences
  - xix. Interleaving String
- 16. Problem Misc
  - i. Nuts and Bolts Problem
- 17. Appendix I Interview and Resume
  - i. Interview
  - ii. Resume

# Data Structure and Algorithm/leetcode/lintcode

[build](#) [passing](#) [Slack](#) [ds-algo](#)

## Introduction

This work is some notes of learning and practicing data structures and algorithm.

1. Part I is some brief introduction of basic data structures and algorithm, such as, linked lists, stack, queues, trees, sorting and etc.
2. Part II is the analysis and summary of programming problems, and most of the programming problems come from <https://leetcode.com/> and <http://www.lintcode.com/>.
3. Part III is the appendix of resume and other supplements.

This project is hosted on <https://github.com/billryan/algorith-exercise> and rendered by [Gitbook](#). You can star the repository on the GitHub to keep track of updates. Another choice is to subscribe channel `#github_commit` via Slack [https://ds-algo.slack.com/messages/github\\_commit/](https://ds-algo.slack.com/messages/github_commit/). [RSS feed is under development](#).

Feel free to access <http://slackin4ds-algo.herokuapp.com> for Slack invite automation.

You can view/search this document online or offline, feel free to read it. :)

- Online(Rendered by Gitbook): <http://algorithm.yuanbin.me>
- Offline(Compiled by Gitbook and Travis-CI):
  1. EPUB: [GitHub](#), [Gitbook](#), [GitCafe\(mainland China\)](#) - Recommended for iPhone/iPad/MAC
  2. PDF: [GitHub](#), [Gitbook](#), [GitCafe\(mainland China\)](#) - Recommended for Desktop
  3. MOBI: [GitHub](#), [Gitbook](#), [GitCafe\(mainland China\)](#) - Recommended for Kindle
- Site Search via Google: `keywords site:algorithm.yuanbin.me`
- Site Search via Swifttype: Click `Search this site` on the right bottom of webpages

## License

This work is licensed under the **Creative Commons Attribution-ShareAlike 4.0 International License**. To view a copy of this license, please visit <http://creativecommons.org/licenses/by-sa/4.0/>

## Multilanguage

- [English](#) maintained by who?
- [简体中文](#) maintained by [@billryan](#)
- [繁體中文](#) maintained by [@CrossLuna](#)

## How to Contribute

If you find any mistakes or want to update/translate the awesome notes, please follow the [contributing guidelines](#). Any comments or suggestions would be greatly appreciated.

## To Do

- [] add multiple languages support, currently 繁體中文, 简体中文 are available
  - [] explore nice writing style
  - [] add implementations of Python, C++, Java regarding leetcode/lintcode OJ platform
  - [] add time and space complexity analysis
  - [] summary of basic data structure and algorithm
  - [x] add CSS for online website <http://algorithm.yuanbin.me>, yahei plugin works well
  - [x] add proper Chinese fonts for PDF output
- 

## 简介

本文档为数据结构和算法学习笔记，全文大致分为以下三大部分：

1. Part I 为数据结构和算法基础，介绍一些基础的排序/链表/基础算法
2. Part II 为 OJ 上的编程题目实战，按题目的内容分章节编写，主要来源为 <https://leetcode.com/>, <http://www.lintcode.com/>, <http://www.geeksforgeeks.org/>, <http://hihocoder.com/>, <https://www.topcoder.com/>.
3. Part III 为附录部分，包含如何写简历和其他附加材料

本文参考了很多教材和博客，凡参考过的几乎都给出明确链接，如果不小心忘记了，请不要吝惜你的评论和issue :)

你可以在线或者离线查看/搜索本文档，以下方式任选~

- 在线阅读(由 Gitbook 渲染) <http://algorithm.yuanbin.me>
- 离线阅读：推送到GitHub后会触发 travis-ci 的编译，相应的部分编译输出提供 GitHub 和 GitCafe 下载。
  1. EPUB: [GitHub](#), [Gitbook](#), [GitCafe\(中国大陆用户适用\)](#) - 适合在 iPhone/iPad/MAC 上离线查看，实测效果极好。
  2. PDF: [GitHub](#), [Gitbook](#), [GitCafe\(中国大陆用户适用\)](#) - 推荐下载 GitHub 和 GitCafe 的版本，Gitbook 官方使用的中文字体有点问题。
  3. MOBI: [GitHub](#), [Gitbook](#), [GitCafe\(中国大陆用户适用\)](#) - Kindle 专用，未测试，感觉不适合在 Kindle 上看此类书籍，尽管 Kindle 的屏幕对眼睛很好...
- Google 站内搜索: keywords site:[algorithm.yuanbin.me](http://algorithm.yuanbin.me)
- Swiftype 站内搜索: 可使用网页右下方的 Search this site 进行站内搜索

## 订阅更新

本项目托管在 <https://github.com/billryan/algorithm-exercise> 由 Gitbook 渲染生成 HTML 页面。

你可以在 GitHub 中 star 该项目查看更新，也可以订阅 [https://ds-algo.slack.com/messages/github\\_commit/](https://ds-algo.slack.com/messages/github_commit/) 中的 #github\_commit channel 在邮件中查看更新细节，RSS 种子功能正在开发中。

Slack 的自助邀请注册功能已启用，访问 <http://slackin4ds-algo.herokuapp.com> 即刻开启~

## 许可证

本作品采用 [知识共享署名-相同方式共享 4.0 国际许可协议](#) 进行许可。传播此文档时请注意遵循以上许可协议。关于本许可证的更多详情可参考 <http://creativecommons.org/licenses/by-sa/4.0/>

本着独乐乐不如众乐乐的开源精神，我将自己的算法学习笔记公开和小伙伴们讨论，希望高手们不吝赐教。

## 多国语言

- English maintained by who?
- 简体中文 maintained by [@billryan](#)
- 繁體中文 maintained by [@CrossLuna](#)

## 如何贡献

如果你发现任何有错误的地方或是想更新/翻译本文档, 请毫不犹豫地猛击 [贡献指南](#).

## 捐助

添加这一小节其实是有点诚惶诚恐的, 毕竟这本小书目前还很不完善, 把捐助信息贴出来不脸红吗? :-( 但既然前些天有网友专门发邮件来问这个事, 我就大概说下我目前的想法, 先以 @billryan 为例, 其他 Contributor 后续补充, 捐助者后期会单独整理公布, 当然这是在征得捐助人同意的前提下进行的。除了在 GitHub 上协助编写文档外, 你还可以以下几种方式回馈各位贡献者:

### 邮寄明信片

@billryan 喜欢收集各种明信片, 来者不拒~ 邮寄的话可以邮寄至 上海市闵行区上海交通大学闵行校区电院群楼5号楼307 , 这个地址 2016 年3月前有效, 收件人: 袁斌。

### 送书

除了邮寄明信片, 你还可以买本书送给各位贡献者, @billryan 的地址见上节。

### 支付宝打赏



金额随意。

## PayPal

账户名：yuanbin2014(at)gmail.com 付款时选择 friends and family

金额随意。

## To Do

---

- [] 添加多国语言支持(English, 繁體中文, 简体中文)
- [] 探索适合用于后期批处理的书写及排版格式
- [] 完善 leetcode/lintcode 部分 C++, Java, Python 三大语言的实现
- [] 加入时间/空间复杂度分析
- [] Part I 部分数据结构和算法基础知识的总结
- [x] 完善在线版本 <http://algorithm.yuanbin.me> 的 CSS, 使用 yahei 插件初步达到目标
- [x] 完善离线版本如 PDF(适合打印的字型) 的中文支持
- [] 完善离线版本如 PDF(适合在电子屏上浏览的字型) 的中文支持

## Part I - Basics

---

# Data Structure

---