



# 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

---

 GITTER  JOIN CHAT →  build  passing

## 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/algorithm-exercise> and rendered by [Gitbook](#). You can star the repository on the GitHub to keep track of updates. RSS feed is under development.

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. [Gitbook](#) - Recommended for iPhone/iPad/MAC
  2. PDF. [Gitbook](#) - Recommended for Desktop
  3. MOBI. [Gitbook](#) - 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/>

## How to Contribute

---

If you find any mistakes or want to update/translate the awesome notes, please follow the [contributing guidelines](#).

## To Do

---

- [ ] add multiple languages support, currently only 繁體中文, 简体中文 are available, English is under construction
- [ ] 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

## Part I - Basics

---

# Data Structure

---