

LeetCode 刷题笔记 V2.1

前言

- 版权归属：九章算法（杭州）科技有限公司
- 可以原文转载和分享，转载时需保留此版权信息，不得对内容进行增删和修改
- 本文作者：九章算法助教团队
- 官方网站：https://www.lintcode.com/?utm_source=sc-libao-thx

背景介绍

本资料包含以下内容：

- 算法和数据结构知识点的学习难度、必刷题目
- 算法面试中高频知识点及考察频率
- LeetCode/LintCode 刷题实用技巧

目录

LeetCode 刷题笔记 V2.1.....	1
前言.....	1
一、 算法面试高频必刷 20 题.....	2
二、 算法和数据结构考察情况.....	2
三、 LeetCode/LintCode 刷题技巧.....	13

一、算法面试高频必刷 20 题

1. [三数之和](#)
2. [子集](#)
3. [第 K 大元素](#)
4. [数组划分](#)
5. [木材加工](#)
6. [最多有 k 个不同字符的最长子字符串](#)
7. [搜索旋转排序数组](#)
8. [最长回文子串](#)
9. [LRU 缓存策略](#)
10. [背包问题](#)
11. [岛屿的个数](#)
12. [验证二叉查找树](#)
13. [有效回文串](#)
14. [单词接龙](#)
15. [最长上升子序列](#)
16. [颜色分类](#)
17. [图是否是树](#)
18. [骑士的最短路线](#)
19. [数字三角形](#)
20. [跳跃游戏](#)

二、算法和数据结构考察情况

1、字符串处理 String

考察情况

考得很多，主要注重代码实现能力，算法上没有太多难点，通常是处理麻烦。

学习难度：★★

最少刷题数：20

必刷题目：

1. [字符串查找](#)

2. [最长回文子串](#)
3. [有效回文串](#)
4. [单词拆分 I](#)
5. [正则表达式匹配](#)
6. [旋转字符串](#)
7. [解码方法](#)
8. [至少 K 个不同字符的子串](#)
9. [字符串查找 II](#)
10. [全零子串的数量](#)
11. [最长公共子序列](#)
12. [编辑距离](#)
13. [第一个只出现一次的字符](#)
14. [乱序字符串](#)
15. [Excel 表列标题](#)
16. [一次编辑距离](#)
17. [删除无效的括号](#)
18. [单词的添加与查找](#)
19. [基础计算器](#)
20. [比较字符串](#)

2、排序算法 Sorting

考察情况

直接考得很少，一般是考察其中的快速排序和归并排序及相关的题，必须背诵这两个。

学习难度：★★

最少刷题数：2

必刷题目：

[字符大小写排序](#)

[错位词分组](#)

3、双指针算法 Two Pointers

考察情况

高频算法之王，变形特别多，算法不算特别难，但能快速想到和写好不容易。

学习难度：★★★★

最少刷题数：20

必刷题目：

1. [颜色分类](#)
2. [两数之和 III-数据结构设计](#)
3. [排颜色](#)
4. [最长子串覆盖](#)
5. [有效回文](#)
6. [带环链表](#)
7. [交错正负数](#)
8. [最接近的三数之和](#)
9. [四数之和](#)
10. [接雨水](#)
11. [寻找重复的数](#)
12. [字符大小写排序](#)
13. [子串字谜](#)
14. [检查缩写字](#)
15. [奇偶分割数组](#)
16. [三数之和小于](#)
17. [移动车棚](#)
18. [粉刷天花板](#)
19. [装最多水的容器](#)
20. [目标移动](#)

4、二分法 Binary Search

考察情况

考察频率中等，能写好写对不容易，二分答案的问题甚至很难想到算法，要背模板。

学习难度：★★★★

最少刷题数：10

必刷题目：

1. [经典二分查找问题](#)
2. [搜索旋转排序数组](#)
3. [寻找峰值](#)
4. [最长上升子序列](#)
5. [木材加工](#)
6. [书籍复印](#)
7. [加热器](#)
8. [地图跳跃](#)
9. [寻找单词](#)
10. [分享巧克力](#)

5、分治法 Divide&Conquer

考察情况

考察频率中等，一般和二叉树一起出现和考察，题一般不难。

学习难度：★★★

最少刷题数：10

必刷题目：

1. [子集](#)
2. [数组划分](#)
3. [验证二叉查找树](#)
4. [全排列](#)
5. [克隆图](#)
6. [排颜色](#)
7. [子数组之和](#)
8. [LRU 缓存策略](#)
9. [单词接龙](#)
10. [单词拆分 I](#)

6、宽度优先搜索 BFS

考察情况

考察频率高，实现一般都不难。

学习难度：★

最少刷题数：5

必刷题目：

1. [岛屿的个数](#)
2. [序列重构](#)
3. [拓扑排序](#)
4. [课程表](#)
5. [买卖股票的最佳时期](#)

7、深度优先搜索 DFS / 递归 Recursion

考察情况

考察频率高，主要是考递归会不会写。

学习难度：★★★★★

最少刷题数：20

必刷题目：

1. [子集](#)
2. [图是否是树](#)
3. [子数组之和](#)
4. [电话号码的字母组合](#)
5. [K 数和](#)
6. [因式分解](#)
7. [N 皇后问题](#)
8. [子模式 II](#)
9. [寻找丢失的数 II](#)

10. [二叉树的最小深度](#)
11. [最小路径和](#)
12. [数字组合 II](#)
13. [下一个排列](#)
14. [分割字符串](#)
15. [恢复 IP 地址](#)
16. [删除无效的括号](#)
17. [数独](#)
18. [单词矩阵](#)
19. [组合](#)
20. [左叶子的和](#)

8、二叉树的遍历算法 Traversal

考察情况

考察频率中等，最常考中序遍历非递归。

学习难度：★

最少刷题数：5

9、拓扑排序算法 Topological Order

考察情况

考察频率中等，但每个公司基本都有一个这个算法的题。

学习难度：★★★

最少刷题数：3

必刷题目：

1. [序列重构](#)
2. [外星人字典](#)
3. [拓扑排序](#)

10、链表 LinkedList

考察情况

中小公司考得多，大公司近年来考得少，题目一般不难，主要考察 **Reference**。

学习难度：★★

最少刷题数：20

必刷题目：

1. [合并 k 个排序链表](#)
2. [带环链表](#)
3. [翻转链表](#)
4. [两个链表的交叉](#)
5. [链表划分](#)
6. [重排链表](#)
7. [回文链表](#)
8. [加一链表](#)
9. [奇偶链表](#)
10. [两数相乘](#)

11、二叉树 Binary Search

考察情况

中小公司考得多，大公司近年来考得少，题目一般不难，主要考察 **Reference**。

学习难度：★★

最少刷题数：20

必刷题目：

1. [二叉树的后序遍历](#)

2. [二叉树的所有路径](#)
3. [根据前序和后序遍历构造二叉树](#)
4. [对称树](#)
5. [找出树中每行的最大值](#)
6. [寻找重复的子树](#)
7. [BST 中第 K 小的元素](#)
8. [具有最大平均数的子树](#)
9. [最小路径和](#)
10. [二叉树的右视图](#)

12、堆 Heap

考察情况

高频，经常会用到，原理必须掌握，但不用掌握代码实现，应用必须掌握代码。

学习难度：★★★

最少刷题数：5

必刷题目：

1. [丑数](#)
2. [堆化](#)
3. [滑动窗口的中位数](#)
4. [大楼轮廓](#)
5. [超级丑数](#)

13、哈希表 Hash Table

考察情况

高频，应用和原理都需要掌握且需要掌握代码实现。

学习难度：★★★

最少刷题数：10

必刷题目：

1. [两数之和](#)
2. [最长回文串](#)
3. [两数组的交集](#)
4. [数据流中第一个唯一的数字](#)
5. [两数和 II-输入已排序的数组](#)
6. [单词接龙](#)
7. [有效回文串](#)
8. [尾部的零](#)
9. [跳跃游戏](#)
10. [数字三角形](#)

14、线段树 Segment Tree

考察情况

不太考，有的题目存在多种解法的时候，线段树可以帮忙降低思考难度。

学习难度：★★★

最少刷题数：3

必刷题目：

[线段树的查询](#)

[线段树的构造](#)

[约翰的生意](#)

15、树状数组 Binary Indexed Tree

考察情况

不太考，与其学这个不如学线段树。

学习难度：★★★

最少刷题数：2

必刷题目：

[统计前面比自己小的数的个数](#)
[可变范围求和](#)

16、跳跃表 Skip List

考察情况

不太考，需要大致知道原理，分布式数据库里会用到这个数据结构。

学习难度：★★★★★

最少刷题数：1

17、字典树 Trie

考察情况

考察频率中等，跟单词有关的问题一般多多少少都可以用到去优化，可替代哈希表。

学习难度：★★★

最少刷题数：3

必刷题目：

1. [子集](#)
2. [三数之和](#)
3. [木材加工](#)

18、并查集 Union Find

考察情况

考察频率中等，主要是 G/F 可能会考，不会的话很多时候可以用 BFS 替代。

学习难度：★★★

最少刷题数：3

必刷题目：

1. [账户合并](#)
2. [打砖块](#)
3. [团队通知](#)

19、动态规划 Dynamic Programming

考察情况

国内大厂基本都考，北美主要是 G/F 喜欢考，其他公司考得少。

学习难度：★★★★★

最少刷题数：50

必刷题目：

1. [栅栏染色](#)
2. [爬楼梯](#)
3. [约翰的后花园](#)
4. [单词拆分](#)
5. [书籍复印](#)
6. [解码方法](#)

- 7. [通配符匹配](#)
- 8. [旅行商问题](#)
- 9. [青蛙跳](#)
- 10. [骰子求和](#)

20、红黑树 Union Find

考察情况

国外只有 G 可能会问到，也只是问大致原理，能干啥，Java 会用 TreeMap 就行。

学习难度：★★★★★★

最少刷题数：1

三、LeetCode/LintCode 刷题技巧

- 如果是天赋极差的选手，前 300 题按着脑袋直接抄，抄多了，你就慢慢走出来了。虽然这是个笨方法，但十分好用。
- 每道题要有 time constraint。大部分 oj 的题目如果需要想+写超过半小时，那就等于完全不会。所以每道题必须掐时间做，做不出来直接看答案，之后码一遍 AC 的代码。
- 除了题目里面的算法内容，刷题还是需要注意基本编程规范和代码质量的，一个是因为面试里面会问到，一个是因为这对你将来真正实战很有帮助。
- 坚持保持手熟很重要，保持题感，这和高考是一模一样的。
- 刷题不需要刷那么多！刷熟常用 200 题刷 5 遍效果>>>>>刷 1000+题
- Talk out loud。这个不仅仅是说在面试的时候你要说出你的思路，刷题的时候也是！到面试的时候才意识到沟通不畅就太晚了。
- 题是永远刷不完的。如果你只刷了 200 题、甚至 100 题、甚至 50 题且只有一遍，个人认为：你也可以去面试。因为：面试这个东西排第一的是运气，是面试官对你的感觉，自信最重要！！但也不要拿你的 dream company 去试水，当炮灰。

