左程云算法小总结

**2022年3月31日**

**版本控制信息**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订人** | **修订时间** | **修订内容** |
| V1.0.0 | 李盼 | 2022/3/22 | 新建 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 基础方法论 5](#_Toc99908653)

[2. 基础算法 5](#_Toc99908654)

[2.1 排序算法 5](#_Toc99908655)

[2.1.1 随机快排及其改造 5](#_Toc99908656)

[2.1.2 堆排序算法 5](#_Toc99908657)

[2.1.3 归并排序 5](#_Toc99908658)

[2.1.4 基数排序 5](#_Toc99908659)

[2.1.5 各种算法的比较 5](#_Toc99908660)

[2.2 链表 6](#_Toc99908661)

[2.2.1 链表中是否存在环 6](#_Toc99908662)

[2.2.2 翻转链表 6](#_Toc99908663)

[2.3 二叉树 6](#_Toc99908664)

[2.3.1 二叉树基础操作 6](#_Toc99908665)

[2.3.2 二叉树的递归套路 6](#_Toc99908666)

[2.4 图算法 6](#_Toc99908667)

[2.4.1 生成图 6](#_Toc99908668)

[2.4.2 深度优先遍历 6](#_Toc99908669)

[2.4.3 广度有点遍历 6](#_Toc99908670)

[2.4.4 拓扑排序 6](#_Toc99908671)

[2.4.5 最小生成树 7](#_Toc99908672)

[2.5 从暴力尝试到动态规划 7](#_Toc99908673)

[2.5.1 几种尝试模型及其应用 7](#_Toc99908674)

[2.6 有序表 7](#_Toc99908675)

[2.6.1 平衡二叉树 7](#_Toc99908676)

[2.6.2 AVL树及SB树 7](#_Toc99908677)

[2.6.3 跳表实现 7](#_Toc99908678)

[3. 进阶算法 7](#_Toc99908679)

[3.1 单调栈 7](#_Toc99908680)

[3.2 滑动窗口 7](#_Toc99908681)

[3.3 KMP算法 7](#_Toc99908682)

[3.4 马拉车算法 7](#_Toc99908683)

[3.5 并查集 7](#_Toc99908684)

[3.6 前缀树 7](#_Toc99908685)

[3.7 Morris遍历及其应用 8](#_Toc99908686)

[3.8 LRU（经典算法，必须能手写） 8](#_Toc99908687)

[3.9 LFU 9](#_Toc99908688)

[4. 经典面试题 9](#_Toc99908689)

[4.1 字符串系列 9](#_Toc99908690)

[4.1.1 最长递增子序列 9](#_Toc99908691)

[4.1.2 最长公共子序列 9](#_Toc99908692)

[4.1.3 最小包含子序列 9](#_Toc99908693)

[4.1.4 最长回文子序列 9](#_Toc99908694)

[4.2 买卖股票问题系列 9](#_Toc99908695)

[4.3 水王问题系列 9](#_Toc99908696)

[5. 纯编码技巧问题 10](#_Toc99908697)

[6. 一些烂题 10](#_Toc99908698)

[7. 变态算法（了解即可） 10](#_Toc99908699)

[7.1 后缀数组 10](#_Toc99908700)

[7.1.1 DC3算法 10](#_Toc99908701)

[7.2 线段树 10](#_Toc99908702)

[8. 直接记忆的算法 10](#_Toc99908703)

[8.1 最大公约数 10](#_Toc99908704)

# 基础方法论

时间复杂度

空间复杂度

# 基础算法

## 排序算法

### 随机快排及其改造

### 堆排序算法

### 归并排序

### 基数排序

### 各种算法的比较

时间复杂度/空间复杂度/是否稳定

## 链表

### 链表中是否存在环

### 翻转链表

## 二叉树

### 二叉树基础操作

### 二叉树的递归套路

## 图算法

### 生成图

### 深度优先遍历

### 广度有点遍历

### 拓扑排序

相关代码

com.zaqbest.study.foundation.alg.zcy.s40\_leetcode.top\_interview.topinterviewquestions.Problem\_0207\_CourseSchedule

### 最小生成树

## 从暴力尝试到动态规划

### 几种尝试模型及其应用

## 有序表

### 平衡二叉树

### AVL树及SB树

### 跳表实现

# 进阶算法

## 单调栈

## 滑动窗口

## KMP算法

## 马拉车算法

## 并查集

## 前缀树

前缀树经典实现

## Morris遍历及其应用

文本

描述已自动生成

## LRU（经典算法，必须能手写）

手机屏幕的截图

描述已自动生成

**相关代码**

com.zaqbest.study.foundation.alg.zcy.top\_interview.topinterviewquestions.Problem\_0146\_LRUCache

## LFU

# 经典面试题

## 字符串系列

### 最长递增子序列

### 最长公共子序列

### 最小包含子序列

### 最长回文子序列

## 买卖股票问题系列

买卖股票问题5个问题

## 水王问题系列

问题原型：一个人发帖的数量超过1/2，找到这个人

相关代码

com.zaqbest.study.foundation.alg.zcy.top\_interview.topinterviewquestions.Problem\_0169\_MajorityElement

com.zaqbest.study.foundation.alg.zcy.training.zdemo.term04.class03.Code05\_FindKMajority

# 纯编码技巧问题

# 一些烂题

# 变态算法（了解即可）

## 后缀数组

### DC3算法

## 线段树

# 直接记忆的算法

## 最大公约数

//*保证初始调用的时候，a和b不等于0*

public static int gcd(int a, int b) {  
 return b == 0 ? a : *gcd*(b, a % b);  
}