

[数据结构与算法之美](#)[首页](#) | [🔍](#)

## 春节7天练 | Day 7 : 贪心、分治、回溯和动态规划

2019-02-10 王争



朗读：修阳

时长00:34 大小540.56K



你好，我是王争。今天是节后的第一个工作日，也是我们“春节七天练”的最后一篇。

## 几种算法思想必知必会的代码实现

### 回溯

利用回溯算法求解八皇后问题

利用回溯算法求解 0-1 背包问题

### 分治

利用分治算法求一组数据的逆序对个数

### 动态规划

0-1 背包问题

最小路径和（详细可看 @Smallfly 整理的 Minimum Path Sum）

编程实现莱文斯坦最短编辑距离

编程实现查找两个字符串的最长公共子序列

编程实现一个数据序列的最长递增子序列

### 对应的 LeetCode 练习题（@Smallfly 整理）

Regular Expression Matching（正则表达式匹配）

英文版：<https://leetcode.com/problems/regular-expression-matching/>

中文版：<https://leetcode-cn.com/problems/regular-expression-matching/>

Minimum Path Sum ( 最小路径和 )

英文版：<https://leetcode.com/problems/minimum-path-sum/>

中文版：<https://leetcode-cn.com/problems/minimum-path-sum/>

Coin Change ( 零钱兑换 )

英文版：<https://leetcode.com/problems/coin-change/>

中文版：<https://leetcode-cn.com/problems/coin-change/>

Best Time to Buy and Sell Stock ( 买卖股票的最佳时机 )

英文版：<https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>

中文版：<https://leetcode-cn.com/problems/best-time-to-buy-and-sell-stock/>

Maximum Product Subarray ( 乘积最大子序列 )

英文版：<https://leetcode.com/problems/maximum-product-subarray/>

中文版：<https://leetcode-cn.com/problems/maximum-product-subarray/>

Triangle ( 三角形最小路径和 )

英文版：<https://leetcode.com/problems/triangle/>

中文版：<https://leetcode-cn.com/problems/triangle/>

---

到此为止，七天的练习就结束了。这些题目都是我精选出来的，是基础数据结构和算法中最核心的内容。建议你一定要全部手写练习。如果一遍搞不定，你可以结合前面的章节，多看几遍，反复练习，直到能够全部搞定为止。

学习数据结构和算法，最好的方法就是练习和实践。我相信这在任何知识的学习过程中都适用。

最后，祝你工作顺利！学业进步！



# 数据结构与算法之美

为工程师量身打造的数据结构与算法私教课

王争

前 Google 工程师



新版升级：点击「👤请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得转载

上一篇 春节7天练 | Day 6：图

下一篇 用户故事 | Jerry银银：这一年我的脑海里只有算法

精选留言 (19)

写留言

**kai**

2019-02-11

3

听了老师的课程，第一遍的时候，只是在读，现在开始回顾：

课程相关的知识点，做了笔记：

<https://github.com/guokaide/algorithm/blob/master/summary/algorithm.md>

课程涉及的题目，也在逐步总结当中：

<https://github.com/guokaide/algorithm/blob/master/questions/questions.md...>

展开 ▾

**kai**

2019-02-11

2

8皇后问题

```
public class EightQueen {
```

```
    private static final int QUEEN_NUMBER = 8; // 皇后个数...
```

展开 ▾

**kai**

2019-02-11

2

动态规划，感觉是面试必考内容，今天跟着这些题目再来复习一遍~

**李皮皮皮皮**

2019-02-11

1

每天一道算法题，风雨无阻（过年偷懒不算😏）

---



**Richard**

2019-02-11

👍 1

老师留的题都很不错，正在刷之前没做过的LeetCode题。

参与下答对三题送课程的活动：

Day 1：

1.求众数(Python)

class Solution:

```
    def majorityElement(self, nums):  
        return sorted(nums)[len(nums) // 2]
```

2.缺失的第一个正数(Golang)

```
func firstMissingPositive(nums []int) int {  
    if len(nums) == 0 {  
        return 1  
    }  
  
    var arr = make([]bool, len(nums)+1)  
    var idx = 1  
    for i := 0; i < len(nums); i++ {  
        if nums[i] >= 0 && nums[i] < len(arr) {  
            arr[nums[i]] = true  
        }  
    }  
  
    for i := 1; i < len(arr); i++ {  
        if arr[i] == false {
```

```
            return i  
        }  
    }  
    return len(arr)
```

```
    for i := 1; i < len(arr); i++ {  
        if arr[i] == false {
```

```
        idx = i
        break
    } else {
        idx = i + 1
    }
}
```

```
return idx
}
```

Day 7:

### 3. 买卖股票的最佳时机(Python)

class Solution:

```
def maxProfit(self, prices):
    if not prices:
        return 0
    min_price = prices[0]
    res = 0
    for i in prices[1:]:
        min_price = min(min_price, i)
        if res < i - min_price:
            res = i - min_price
    return res
```



\_CountingSta...

2019-02-11

买卖股票的最佳时机 go 语言实现  
package main





```
import "fmt"
```

...

展开 ∨

---



纯洁的憎恶

2019-02-10

👍 1

这冲刺压力有点大了 😞

---



吴...

2019-02-14

👍

回溯0-1背包问题

```
#include<iostream>
```

```
using namespace std;
```

```
int v[10] = {2,2,4,6,3};
```

```
int M;//代表背包的容积 ...
```

展开 ∨

---



吴...

2019-02-14

👍

```
#include<iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
int queen[100];...
```

展开 ∨

**纯洁的憎恶**

2019-02-12



第一题，把39讲的代码改了一下。。。

```
public class Pattern {  
    private boolean matched = false;  
    private char[] pattern; // 正则表达式...
```

展开 ∨

**卡罗**

2019-02-12



感觉，明天就是专栏的最后一天

**黄丹**

2019-02-11



课程的最后一天，也是新年上班的第一天，感谢王老师的教育和陪伴，祝您生活开心，工作顺利。

今天的题目比前几天的都难一点，只做了三题，太累了TaT。对于动态规划和贪心总觉得很巧妙，如果想不到动态转移方程式，就很难做，但要是想到了，真的是豁然开朗。对于这一类题，还是要多锻炼，找动态转移方程式要从最后一个结果出发，去想这个结果可以由什么...

展开 ∨

**C\_love**

2019-02-11



Best Time to Buy and Sell Stoc...

```
class Solution {  
    public int maxProfit(int[] prices) {  
        if (prices == null || prices.length == 0 || prices.length == 1) {...
```

展开 ∨

**ext4**

2019-02-11



最小路径和

```
class Solution {  
    public:  
        int minPathSum(vector< vector<int> >& grid) {  
            int m = grid.size();...
```

展开 ∨

**Kyle Liu**

2019-02-11



有兴趣朋友可以将思路分析提交到<https://github.com/kylesliu/awesome-golang-leetcode>，欢迎大家提issue

**幻月剑**

2019-02-11



一刷完成，题目没有做，算是草草学了一遍，等第二遍将题目都做完

---

**峰**

2019-02-11



七七八八跟着老师复习了遍算法，印象最深刻的无疑是老师结合具体的应用场景讲利用的数据结构与算法！棒棒哒

---

**小美**

2019-02-10



这期题目好难，而且在Web网页上找对应的课程有点麻烦，能否在每个Day的文章中加上对应文章的链接呢？

---

**虎虎**

2019-02-10



正则表达式

```
public boolean isMatch(String s, String p) {  
  
    if (s == null || p == null) {  
        return false;  
    }  
    boolean[][] dp = new boolean[s.length()+1][p.length()+1];  
    dp[0][0] = true;
```

```
for (int i = 0; i < p.length(); i++) {
    if (p.charAt(i) == '*' && dp[0][i-1]) {
        dp[0][i+1] = true;
    }
}
for (int i = 0 ; i < s.length(); i++) {
    for (int j = 0; j < p.length(); j++) {
        if (p.charAt(j) == '.') {
            dp[i+1][j+1] = dp[i][j];
        }
        if (p.charAt(j) == s.charAt(i)) {
            dp[i+1][j+1] = dp[i][j];
        }
        if (p.charAt(j) == '*') {
            if (p.charAt(j-1) != s.charAt(i) && p.charAt(j-1) != '.') {
                dp[i+1][j+1] = dp[i+1][j-1];
            } else {
                dp[i+1][j+1] = (dp[i+1][j] || dp[i][j+1] || dp[i+1][j-1]);
            }
        }
    }
}
return dp[s.length()][p.length()];
}
```

leetcode的排名第一的答案，搬过来了

