

## 春节7天练 | Day 2: 栈、队列和递归

2019-02-05 王争



朗读：修阳

时长00:50 大小803.06K



你好，我是王争。初二好！

为了帮你巩固所学，真正掌握数据结构和算法，我整理了数据结构和算法中，必知必会的30个代码实现，分7天发布出来，供你复习巩固所用。今天是第二篇。

和昨天一样，你可以花一点时间，来完成测验。测验完成后，你可以根据结果，回到相应章节，有针对性地进行复习。

### 关于栈、队列和递归的几个必知必会的代码实现

#### 栈

用数组实现一个顺序栈

用链表实现一个链式栈

编程模拟实现一个浏览器的前进、后退功能

## 队列

用数组实现一个顺序队列

用链表实现一个链式队列

实现一个循环队列

## 递归

编程实现斐波那契数列求值  $f(n)=f(n-1)+f(n-2)$

编程实现求阶乘  $n!$

编程实现一组数据集合的全排列

## 对应的 LeetCode 练习题 (@Smallfly 整理)

### 栈

Valid Parentheses (有效的括号)

英文版: <https://leetcode.com/problems/valid-parentheses/>

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

Longest Valid Parentheses (最长有效的括号)

英文版: <https://leetcode.com/problems/longest-valid-parentheses/>

中文版: <https://leetcode-cn.com/problems/longest-valid-parentheses/>

Evaluate Reverse Polish Notation (逆波兰表达式求值)

英文版: <https://leetcode.com/problems/evaluate-reverse-polish-notation/>

中文版: <https://leetcode-cn.com/problems/evaluate-reverse-polish-notation/>

## 队列

Design Circular Deque (设计一个双端队列)

英文版: <https://leetcode.com/problems/design-circular-deque/>

中文版: <https://leetcode-cn.com/problems/design-circular-deque/>

Sliding Window Maximum (滑动窗口最大值)

英文版: <https://leetcode.com/problems/sliding-window-maximum/>

中文版: <https://leetcode-cn.com/problems/sliding-window-maximum/>

## 递归

Climbing Stairs (爬楼梯)

英文版: <https://leetcode.com/problems/climbing-stairs/>

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

---

昨天的第一篇, 是关于数组和链表的, 如果你错过了, 点击文末的“[上一篇](#)”, 即可进入测试。

祝你取得好成绩! 明天见!



# 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 1: 数组和链表

下一篇 春节7天练 | Day 3: 排序和二分查找

## 精选留言 (27)

写留言



ALAN

2019-02-08

import java.util.Arrays;

/\*\*

\*

\*Stack 1 solution...

展开 ▼

1



吴...

2019-02-06

之前有个类似的题，走楼梯，装苹果，就是把苹果装入盘子，可以分为有一个盘子为空

1

(递归)，和全部装满没有空的情况，找出状态方程，递归就可以列出来了。我觉得最关键是要列出状态方程，之前老师类似于说的不需要关注特别细节，不要想把每一步都要想明白，快速排序与递归排序之类的算法，之前总是想把很细节的弄懂，却发现理解有困难。

**李皮皮皮皮**

2019-02-05

👍 1

基础数据结构和算法是基石，灵活运用是解题的关键。栈，队列这些数据结构说到底就是给顺序表添加约束，更便于解决某一类问题。学习中培养算法的设计思想是非常关键的。而且思想是可以通用的。之前读《暗时间》一书，收获颇深。书中介绍之正推反推我在做程序题时竟出奇的好用。

**yingyingqi...**

2019-02-09

👍

全排列 C++实现

```
void digui(vector<int> res, int i,vector<int> cures)
{
    if (i == res.size())
    {...
```

展开 ▼

**神盾局闹别扭**

2019-02-09

👍

全排列实现：

```
void Dopermute(char *pstr, char *pBegin)
{
    if (*pBegin == '\0')
        printf("%s\n", pstr);...
```

展开 ▼

**molybdenum**

2019-02-09

👍

老师新年好 这是我的作业

[https://blog.csdn.net/github\\_38313296/article/details/86819684](https://blog.csdn.net/github_38313296/article/details/86819684)



你看起来很好...

2019-02-09



爬楼梯python代码实现，需要使用散列表存储已经计算过的数字，这样可以降低时间复杂度，否则Leetcode会报超时错误：

```
class Solution:
    def __init__(self):
        self.buf = {1:1, 2:2}...
```

展开 ▾



你看起来很好...

2019-02-09



设计双端队列python代码：

```
class MyCircularDeque:

    def __init__(self, k: 'int'):
        self.data = [-1] * k...
```

展开 ▾



你看起来很好...

2019-02-09



逆波兰表达式python实现，时间复杂度 $O(n)$ ，空间复杂度 $O(1)$ ，

```
class Solution:
    def evalRPN(self, tokens: 'List[str]') -> 'int':
        data = []...
```

展开 ▾



纯洁的憎恶

2019-02-09



1.维护一个栈，顺序遍历括号序列，若与栈首括号匹配成功，则出栈并遍历下一个括号。遍历完毕后若栈为空则返回true。

2.我比较笨，用空间降低逻辑复杂度吧。申请长度为 $n$ 的bool数组 $S$ ，初始化全为false，记录匹配成功的情况。遍历括号字符串 $A$ ，若当前字符与栈首对应的字符不匹配，或栈为...

展开 ▾





老杨同志  
2019-02-08



```
package com.jxyang.test.geek.day2;
//爬梯子、斐波那契数列
class Solution {
    public int climbStairs(int n) {
        if(n<=0){...
```

展开 ▾



老杨同志  
2019-02-08



全排列

```
import java.util.ArrayList;
import java.util.List;
```

//全排列...

展开 ▾



你看起来很好...  
2019-02-08



有效的括号python代码实现：

```
class Solution:
    def isValid(self, s):
        """
        :type s: str...
```

展开 ▾



老杨同志  
2019-02-06



```
package com.jxyang.test.geek.day2;
//链表实现栈
public class LinkStock<T> {
    private Node<T> head;
    public static void main(String[] args) {...
```

展开 ▾



老杨同志





2019-02-06

```
import java.util.ArrayList;
```

```
//数组实现队列,固定大小, 暂时未实现扩容
```

```
public class ArrayQueue<T> {  
    private Object[] arr;...
```

展开 ▾



失火的夏天

2019-02-06



自己手动实现一个双端队列，其实只要会自己写实现一个链表，思路基本是一致的。用好头尾指针就可以解决一切问题，因为代码太长，就只贴上核心部分了，

```
// 双端队列
```

```
private static class DequeNode{  
    int val;...
```

展开 ▾



失火的夏天

2019-02-06



```
// 有效的括号
```

```
public boolean isValid(String s) {  
    Map<Character,Character> map = new HashMap<>();  
    map.put(')','(');  
    map.put('}','{');...
```

展开 ▾



黄丹

2019-02-06



王争老师新年快乐呀，我今天走亲戚去啦，队列的两题还没做TaT。下面放上栈和递归的四题的解题思路和代码

栈是一种受限制的线性表，只允许在栈顶进行操作（插入，取出，取值），Java已经为我们封装了一个这样的数据结构Stack，对应的函数是（push,pop,peak）

1. Valid Parentheses（有效的括号）...

展开 ▾



\_CountingSta...

2019-02-06





阶乘 go 语言实现

```
package main
```

```
import "fmt"
```

```
...
```

展开 ▾



ext4

2019-02-06



有效的括号

```
class Solution {
```

```
public:
```

```
    bool isValid(string s) {
```

```
        if (s.length() == 0) {...
```

展开 ▾



峰

2019-02-06



第二题，最长有效括号，首先一遍遍历，得到每个(匹配的)的下标，没有匹配的为0，然后对这个中间结果去看最长一段中间不含0的即可， $O(n)$ 的时空复杂度。



虎虎❤️

2019-02-06



循环队列

```
public class LoopQueue<E> implements Queue<E> {
```

```
    private E[] data;
```

```
    private int front, tail;...
```

展开 ▾



C\_love

2019-02-06



Longest Valid Parentheses

```
class Solution {
```

```
    /**
```

\* Time and space complexity:  $O(n)$ ...

展开 ▾



吴...

2019-02-06



个人理解链表与数组都是数据结构基础中的基础，许多复杂的数据结构图，散列表跳表这些数据结构在老师的讲解之下这些数据结构一个个更加形象起来，再加上老师应用的场景，更好的理解。



C\_love

2019-02-06



```
class Solution {  
    private static final Map<Character, Character> PAIRS = new HashMap<>(3);  
    static {  
        PAIRS.put('(', ')');  
        PAIRS.put('[', ']');...  
    }  
}
```

展开 ▾



SyndromePol...

2019-02-06



求斐波那契数列，当然最经典的算法就是递归，但是递归的效率非常低，因为中间过车会计算大量重复的子节点。在《剑指Offer》一书中，提到了一个自下而上计算的方法。我们知道 $f(0)=0, f(1)=1$ ,再计算 $f(2), f(3)$ 一直到 $f(n)$ 。这样，时间复杂度就是 $O(n)$ 。



失火的夏天

2019-02-05



其实爬楼梯的原理就是斐波拉契数列，代码写出来都是一毛一样的呀