

作者：沉默王二

链接：<https://www.zhihu.com/question/66535555/answer/1799868707>

来源：知乎

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

02、第 2 - 12 个小时，Java 基础入门

1) 基本数据类型

- 8 种基本数据类型 (boolean、char、byte、short、int、long、float、double)
- 整形中 byte、short、int、long 的取值范围
- 单精度和双精度
- [为什么不能使用 “==” 比较浮点数](#)
- [基本类型和包装类型的区别](#)
- 自动拆箱与装箱
- Integer 的缓存机制

2) 操作符

- 算术运算符
- 逻辑运算符
- 比较运算符

3) 流程控制语句

- 条件分支 (if/else/else if、三元运算符、[switch](#))
- 循环或者遍历 (for、while、do-while)
- break 和 continue

4) 包

- 创建包
- 导入包
- 包全名

5) main 方法详解

6) [数组](#)

7) 注释

8) 字符串

- 双引号字符串和 new 字符串对象

- [字符串为什么是不可变的?](#)
- String、StringBuilder 和 StringBuffer 之间的区别
- 字符串拼接的几种方式
- 字符串常量池
- 关于 intern
- [字符串比较 \(== 和 equals\)](#)
- [字符串拆分](#)
- [字符串操作小技巧](#)
- [字符串转整形](#)
- [生成 UUID](#) (多谢知友的提醒, 链接已更新)
- [字符串可以引用传递吗?](#)
- [字符串长度](#)
- [关于 substring](#)

03、第 13 -20 个小时, Java 核心技术

1) 面向对象

- [面向对象和面向过程](#)
- [封装](#)
- [继承](#)
- 多态
- [接口](#)
- [抽象类](#)
- [重写和重载](#)
- 构造方法
- 局部变量、成员变量、静态变量、常量
- [值传递还是引用传递](#)
- [final 关键字](#)
- [this 关键字](#)
- super 关键字
- [不可变对象](#)
- [equals\(\) 和 hashCode\(\)](#)
- [打印 Java 对象](#)
- [NullPointerException](#)

2) 常用工具类

- 字符串相关的工具类
- 日期时间相关的工具类
- [枚举](#)
- [随机数](#)
- 正则表达式
- Apache-commons 工具库

- [Guava 工具库](#)

3) 集合框架

- [ArrayList](#)
- [LinkedList](#)
- [ArrayList 和 LinkedList 之间的区别](#)
- [不可变 List](#)
- [CopyOnWriteArrayList](#)
- [HashMap](#)
- [LinkedHashMap](#)
- [TreeMap](#)
- [ConcurrentHashMap](#)
- [fail-fast](#)

4) [反射机制](#)

5) [异常处理](#)

- [try-with-resource](#)

6) [注解](#)

7) IO 流

- 字符流、字节流
- 输入流、输出流
- 同步、异步
- 阻塞、非阻塞
- BIO、NIO 和 AIO
- [NIO 2.0](#)

8) [序列化](#)

9) [泛型](#)

10) 单元测试

- Junit
- TestNG

11) 编码方式

- ASCII
- Unicode
- UTF-8

- GBK、GB2312
- 如何解决乱码问题

12) 并发编程

- 什么是并发
- 什么是并行
- 什么是线程
- 什么是进程
- 线程的状态
- 线程的优先级
- 创建线程
- 创建线程池
- 什么是线程安全
- 多级缓存和一致性问题
- CPU 时间片和原子性问题
- 指令重排和有序性问题
- 线程安全和内存模型
- happens-before
- 可重入锁
- 阻塞锁
- 乐观锁
- 悲观锁
- 分布式锁
- CAS
- ABA
- 偏向锁
- 轻量级锁
- 重量级锁
- 自旋锁
- 什么是死锁
- 如果避免死锁
- [synchronized](#)
- volatile
- ThreadLocal
- Executors
- CountdownLatch
- Thread
- Runnable
- Callable
- ReentrantLock
- ReentrantReadWriteLock
- Atomic 相关类

13) Java 8 新特性

- [Lambda 表达式](#)
- [Stream 流](#)
- [Optional](#)

14) 源码阅读

- String
- Integer
- ArrayList
- LinkedList
- CopyOnWriteArrayList
- HashMap
- TreeMap
- LinkedHashMap
- ConcurrentHashMap
- CopyOnWriteArrayList

04、第 21 -24 个小时，JVM 和性能优化

1) JVM

- Java 内存结构
- 堆
- 栈
- 垃圾回收
- JVM 参数调优
- Java 对象模型
- HotSpot
- [类加载机制](#)
- 编译和反编译
- 反编译工具
- JIT
- 虚拟机性能监控和故障处理工具（jps、jstack、jmap、jstat、jconsole、javap）

[2\) 性能优化](#)

- 使用单例
- 使用线程池
- 减少上下文切换
- 减小锁粒度
- 数据压缩
- Stream 并行流
- GC 调优

- JVM 内存分配调优
- btrace

05、第 2 天，补充基础知识

1) 设计模式

- 设计模式的六大原则
- 创建型设计模式（单例、抽象工厂、建造者、工厂、原型）
- 结构型设计模式（适配器、桥接、装饰、组合、外观、享元、代理）
- 行为型设计模式（模板方法、命令、迭代器、观察者、中介者、备忘录、解释器、状态、策略、责任链、访问者）
- 单例的七种写法

2) 数据结构和算法

- [简单的数据结构（栈、队列、链表、数组、哈希表）](#)
- [树（二叉树、字典树、平衡树、排序树、B 树、B+ 树、R 树、红黑树、多路树）](#)
- 堆
- 图（拓扑、有向图、无向图）
- 稳定的排序算法（冒泡排序、插入排序、鸡尾酒排序、桶排序、计数排序、归并排序、原地归并排序、二叉排序树排序、鸽巢排序、基数排序、侏儒排序、图书馆排序、块排序）
- 不稳定的排序算法（选择排序、希尔排序、梳排序、堆排序、平滑排序、快速排序、内省排序、耐心排序、Clover 排序）
- 时间复杂度
- 空间复杂度
- 贪心算法
- KMP 算法

3) 操作系统

- Linux 常用命令（find、top、tar、move、grep、tail、netstat、curl、wget、ping、ssh）
- 服务器性能指标（qps、CPU 利用率）
- 进程同步
- 分段和分页
- 虚拟内存和主存

4) 网络安全

- CSRF
- XSS

- SQL 注入
- 加密和解密（对称加密、非对称加密）
- MD5、SHA1、DES、RSA
- DDOS 攻击
- HTTP 和 HTTPS
- SSL
- TLS
- TCP 和 UDP
- Cookie、Session
- CDN
- DNS

5) 数据库

- MySql
- 索引
- 存储过程
- 分库分表
- binlog
- 读写分离
- 数据库缓存（[Redis](#)、[MongoDB](#)）
- 数据库中间件（MyCat）
- 数据库连接池（Durid）

6) 大数据

- 搜索（[Elasticsearch](#)、Solr）
- 流式计算（Storm、Spark、Flink）
- Hadoop

7) 服务器

- Tomcat
- jetty
- Nginx

8) 框架

- Spring
- MyBatis
- Spring MVC
- Spring Boot
- Spring Security
- Spring Cloud
- Netty

- Dubbo

9) 消息队列

- [RabbitMQ](#)
- Kafka

10) 容器

- Docker
- K8s

06、第 3 天

前半天刷面试题和算法题，听说 GitHub 上有一份星标 10W+ 的 Java 面试刷题攻略，用量子读书法快速过一遍。

[GitHub 标星 10W+, 有了这份 Java 面试攻略, 和面试官硬刚没毛病](#)



mp.weixin.qq.com/s/MWHDLb8V6ulHPPrQaxBl6Q

还有这份 LeetCode 刷题笔记，整整 300 多道，关键是纯 Java 版的，对于我们 Java 程序员来说，用自己熟悉的编程语言真的能提高刷题的幸福指数。想去大厂的话，算法是必须刷的。

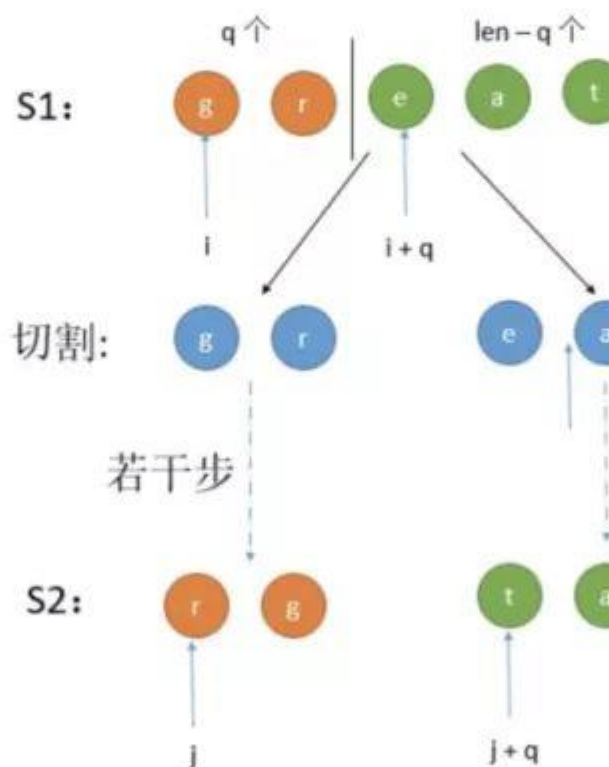
- 70. Climbing Stairs
- 71. Simplify Path
- 72. Edit Distance
- 73. Set Matrix Zeroes
- 74. Search a 2D Matrix
- 75. Sort Colors
- 76. Minimum Window Substring
- 77. Combinations
- 78. Subsets
- 79. Word Search
- 80. Remove Duplicates from Sorted ...
- 81. Search in Rotated Sorted Array II
- 82. Remove Duplicates from Sorted ...
- 83. Remove Duplicates from Sorted ...
- 84. Largest Rectangle in Histogram
- 85. Maximal Rectangle
- 86. Partition List
- 87. Scramble String
- 88. Merge Sorted Array
- 89. Gray Code
- 90. Subsets II
- 91. Decode Ways
- 92. Reverse Linked List II
- 93. Restore IP Addresses
- 94. Binary Tree Inorder Traversal
- 95*. Unique Binary Search Trees II

解法二 动态规划

既然是递归，压栈压栈压栈，出栈出栈出栈，我们可以部往上走。

我们用 $dp[len][i][j]$ 来表示 $s1[i, i + len)$ 和 $s2[j, j + len)$ 是否能转换为 S2 从 j 开始的 len 个字符是否能转换为 S2 从 j 开始的 len 个字符。

第 1 种情况，参考下图：假设左半部分长度是 q ， $dp[j + q]$ 。也就是 S1 的左半部分和 S2 的左半部分以及



第 2 种情况，参考下图：假设左半部分长度是 q ，那么 $len - q][i + q][j]$ 。也就是 S1 的右半部分和 S2 的左



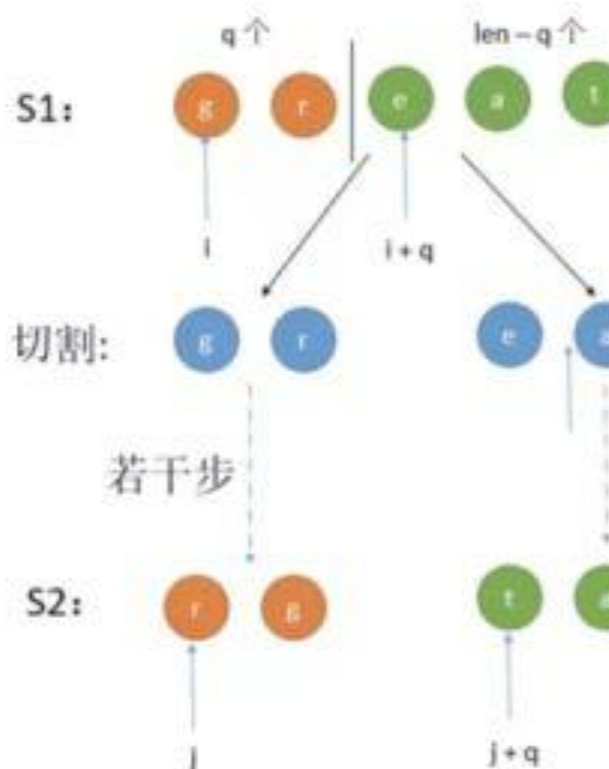
- 70. Climbing Stairs
- 71. Simplify Path
- 72. Edit Distance
- 73. Set Matrix Zeroes
- 74. Search a 2D Matrix
- 75. Sort Colors
- 76. Minimum Window Substring
- 77. Combinations
- 78. Subsets
- 79. Word Search
- 80. Remove Duplicates from Sorted ...
- 81. Search in Rotated Sorted Array II
- 82. Remove Duplicates from Sorted ...
- 83. Remove Duplicates from Sorted ...
- 84. Largest Rectangle in Histogram
- 85. Maximal Rectangle
- 86. Partition List
- 87. Scramble String
- 88. Merge Sorted Array
- 89. Gray Code
- 90. Subsets II
- 91. Decode Ways
- 92. Reverse Linked List II
- 93. Restore IP Addresses
- 94. Binary Tree Inorder Traversal
- 95*. Unique Binary Search Trees II

解法二 动态规划

既然是递归，压栈压栈压栈，出栈出栈出栈，我们可以部往上走。

我们用 $dp[len][i][j]$ 来表示 $s1[i, i + len)$ 和 $s2[j, j + len)$ 从 i 开始的 len 个字符是否能转换为 $S2$ 从 j 开始的 len 个字符。

第 1 种情况，参考下图：假设左半部分长度是 q ， $dp[i][j + q]$ 。也就是 $S1$ 的左半部分和 $S2$ 的左半部分以及



第 2 种情况，参考下图：假设左半部分长度是 q ，那么 $dp[i + q][j]$ 。也就是 $S1$ 的右半部分和 $S2$ 的左



真的是图文并茂，不只是干巴巴的解题代码，可以直接点击领取：[刷完 300 道 LeetCode 题后，我强到可以和面试官硬刚！纯正 Java 版](#)

点击小卡片关注公众号「沉默王二」回复「01」即可获取美团大佬的刷题笔记
后半天准备简历，入职 BAT 大厂！

[入职阿里后，才知道原来简历这么写](#)
mp.weixin.qq.com/s/QFraobvuGnVJRCi4tA-CNQ



“什么？”

“吹牛皮！”

“你说什么？再说一遍？”

“你丫的在吹牛皮！”

“你丫的才吹牛皮呢，信不信老子踹你一脚！”只听咣当一声，我感觉脚指头要断了，原来我踹到墙了！！！！！！！！！！疼啊。

“兄弟，你刚刚在干嘛？没事吧你？醒醒！你吓我一大跳！”睡在我下铺的兄弟，少博关心地问。

“疼疼疼疼。。。。。。别动，疼。。。。。”我摸着脚指头哭着说。