# 算法

算法问题可以分为以下三类：

**基础问题(1)**：即使是新手，一眼看过去就有思路，只是实现的时候需要注意细节。

**普通问题(2)**：这些问题通常属于以上分类中的某一类，需要面试者掌握一些常见的思路，比如递归、动态规划、BFS/DFS、双指针、二分搜索 等。或者是直接考察数据结构的使用，如：哈希、栈和队列、链表等，如果具备了这些基础知识，此类题目通常能够比较快速的解决。

**进阶问题(3)**：这些题的解题思路和普通问题相似，但是需要你事先有对应的知识积累，否则难以直接看出问题的本质。

**疑难杂题(4)**：这类问题比较奇怪，解决它以后并不能给别的题目太多帮助，如果时间紧张可以暂时放弃。

在“【】”中标记了我对此题类型的分类，如果加星号表示此题在实际面试中出现过。

## 字符串

【3】最长回文子串

【3】最长无重复子串

【1\*】字符串转数字

【4】KMP 算法

【2】字符串全排列

【2\*】翻转字符串

## 动态规划

【2】背包问题

【3】连续子数组的最大和

【4】实现简单的正则表达式匹配

## 数组

【3】求两个等长、有序数组的中位数（二分法）

【4】求两个不等长、有序数组的中位数

【3】旋转数组求最小值、【3】旋转数组求查找某个值是否存在（二分法）

【4\*】每行从左到右，每列从上到下递增的二维数组中，判断某个数是否存在（剑指 offer 第 3 题）

【3\*】数组中出现次数超过一半的数字

【3\*】第 k 大的数（拓展：最大的 k 个数）

【3\*】有序数组中某个数字出现的次数（提示：利用二分搜索）

## 链表

【2】反转链表（使用递归和迭代两种解法，了解头插法）

【3】删除链表的当前节点

【3】删除倒数第 k 个节点

【1】两个有序链表合并

【4】复杂链表的复制

【2\*】判断链表是否有环

【3\*】两个链表的第一个公共节点（提示：考虑链表有环的情况）

【3】删除链表中重复节点

## 树

【3】根据中序和后序遍历结果重建二叉树、【3】根据中序和前序遍历结果重建二叉树

【2】翻转二叉树

【2】从上往下打印二叉树 (BFS 的思想)

【3】判断某个数组是不是二叉树的后序遍历结果 (剑指 offer 第 24 题)

【3】二叉树中和为某个值的路径

【3\*】二叉树中某个节点的下一个节点 (强烈推荐准备一下，剑指 offer 第 58 题)

## 栈

【2】用两个栈实现队列、【2】用两个队列实现栈

【2】实现一个栈，可以用常数级时间找出栈中的最小值

【3】判断栈的压栈、弹栈序列是否合法（剑指offer 第 22 题）

## 排序

了解以下排序的时间、空间复杂度，是否稳定，实现原理

归并排序、拓展：求数组中的逆序对个数

快速排序 重点：partion 函数的实现

堆排序

数组元素值域已知时，考虑 基数排序 和 桶排序

位运算

【2】给一个十进制数字，求它的二进制表示中，有多少个 1 (n &= n - 1)

【3】给一个数组，所有数字都出现了偶数次，只有一个出现了一次，找出这个数

【4】给一个数组，所有数字都出现了三次，只有一个出现了一次，找出这个数

【3】给一个数组，所有数组都出现了偶数次，只有两个数字出现了一次，找出这两个数

# 网络

【博客】我的六篇总结 <http://www.jianshu.com/nb/3276500>

1. 简介 TCP 和 UDP 区别，他们位于哪一层？
2. 路由器和交换机的工作原理大概是什么，他们分别用到什么协议，位于哪一层？
3. 描述TCP 协议三次握手，四次释放的过程。
4. TCP 协议是如何进行流量控制，拥塞控制的？
5. 为什么建立连接时是三次握手，两次行不行？如果第三次握手失败了怎么处理
6. 关闭连接时，第四次握手失败怎么处理？
7. 你怎么理解分层和协议？
8. HTTP 请求中的 GET 和 POST 的区别，Session 和 Cookie 的区别。
9. 谈谈你对 HTTP 1.1，2.0 和 HTTPS 的理解。

# 操作系统与编译

推荐阅读《程序员的自我修养》

程序员的自我修养读书笔记：<https://link.jianshu.com/?t=https://bestswifter.com/cheng-xu-yuan-de-zi-wo-xiu-yang-du-shu-bi-ji/>

1. 源代码是怎么变成可执行文件的，每一步的作用是什么？（预编译，词法分析，语法分析，语义分析，中间语言生成目标代码生成，汇编，链接）
2. 应用层、API、运行库、系统调用、操作系统内核之间的关系是什么？
3. 虚拟内存空间是什么，为什么要有虚拟内存空间。
4. 静态链接和动态链接分别表示什么，大概是怎么实现的？
5. 可执行文件的结构如何？（分为哪些段）
6. 它是怎么装载进内存的，为什么要分段，分页，页错误是什么？
7. 进程的内存格局是怎样的？（堆、栈、全局/静态区，代码区，常量区）
8. 堆和栈的区别，函数调用和栈的关系
9. 进程和线程的区别
10. 异步和同步，串行，并发，并行的区别
11. 多并发任务，仅多线程能加快速度么（不能，会变慢，有线程切换的开销）
12. 多个线程之间可以共享那些数据
13. 进程之间如何通信
14. 介绍几种锁，他们的用途和区别