# 数据结构

## 数组

## 链表

## 栈

## 队列

## 树

### 1.二叉树

## 哈希表

## 堆

https://blog.csdn.net/liuyuchen282828/article/details/89255389

### 1.大跟堆

### 2.小跟堆

### 3.可并堆

## 字符串

### 字典树

<https://www.jianshu.com/p/617d8fd5dde6>

1. 查找、插入、删除的时间复杂度都是 **O(len)**

### 后缀树

<https://www.jianshu.com/p/60584ab3729a>

## 跳表

<https://www.jianshu.com/p/9d8296562806>

1查找、插入、删除的时间复杂度都是 **O(logn)**

2空间复杂度**O(n)**

# 算法

## 1.排序算法

### 冒泡

<http://data.biancheng.net/view/116.html>

1时间复杂度：最差O(n2)，最好O(n)，平均O(n2)

2空间复杂度O(1)

3稳定性：稳定

## 选择排序

https://blog.csdn.net/qq\_33289077/article/details/90263964

#### 简单选择排序

<https://zhuanlan.zhihu.com/p/123048793>

1时间复杂度：最差O(n2)，最好O(n2)，平均O(n2)

2空间复杂度O(1)

3稳定性：不稳定

#### 堆排序

<https://zhuanlan.zhihu.com/p/12488505>

1时间复杂度：最差O(nlogn)，最好O(nlogn)，平均O(nlogn)

2空间复杂度O(1)

3稳定性：不稳定

### 插入排序

### 希尔排序

### 快速排序

### 归并排序

### 计数排序

### 桶排序

### 基数排序

## 2.搜索算法

## 二分查找

## 动态规划

## 贪婪算法

# 简历内容写法

### STAR模刑



**Situation:**简短的项目背景。比如项目的规模,开发的软件的功能、目标用户等。

**Task:**自己完成的任务。这个要写详细，要让面试官对自己的工作一目了然。在用词上要注意区分“参与”和“负责”:如果只是加入某一个开发团队写了几行代码就用“负责”，那就很危险。面试官看到简历上应聘者“负责”了某个项目，他可能就会问项目的总体框架设计、核心算法、团队合作等问题。这些问题对于只是简单“参与”的人来说，是很难回答的，会让面试官认为你不诚实，印象分会减去很多。

**Action:**为完成任务自己做了哪些工作，是怎么做的。这里可以详细介绍。做系统设计的，可以介绍系统架构的特点;做软件开发的,可以写基于什么工具在哪个平台下应用了哪些技术;做软件测试的，可以写是手工测试还是自动化测试、是白盒测试还是黑盒测试等。

**Result:**自己的贡献。这方面的信息可以写得具体些，最好能用数字加以说明。如果是参与功能开发,则可以说按时完成了多少功能; 如果做优化，则可以说性能提高的百分比是多少;如果是维护，则可以说修改了多少个Bug。

# 为什么跳槽答法

笔者在面试的时候通常给出的答案是:现在的工作做了一段时间，己经没有太多的激情了，因此希望寻找一份更有挑战的工作。然后具体论述为什么有些厌倦现在的职位，以及面试的职位我为什么会有兴趣。

# 提问环节

## 三不问

不要问和自己职位没有关系的问题、不要问薪水、不要当面问面试结果

## 推荐问

与职位或者项目相关的问题

# 面试官关注的五个方面

基础知识是否扎实、能否写出高质量的代码、思路是否清晰、是否有优化效率的能力，以及包括学习能力、沟通能力在内的综合素质是否优秀。

# 问题

## 斐波那契数列

## 青蛙跳台阶