

2019.04.27 分享答疑

1、自我介绍 (@婷博士)

- 27岁计算机博士毕业，毕业后从事金融风控算法工作三年，目前从事搜索广告算法工作
- 数据科学与数据驱动的践行者
- 研究生学计算机是偶然，也经历过转专业的痛苦，选择了互联网就会坚持下去

2、算法学习经验和学习路径

- 算法分两大块，数据结构算法 和 应用算法
- 数据结构算法是基础，也是敲门砖，但不是工作的全部，可能工作中从来都不要写，但却是一种潜移默化的能力；数据结构的算法学习经验就是“重复、重复、再重复，刷题、刷题、再刷题，思考、思考、再思考”；快排算法非常重要！
- 应用算法是一切与业务相关的算法统称，涉及的算法解决方案可能包括有机器学习、深度学习、数据挖掘、数据科学、大数据技术、工程技术等等；应用算法的学习门槛近些年已经非常低了，就是通常说的“调个包”、“调个框架”，“调个参”，但如果到专家级别，不仅需要扎实的机器学习算法基础，还需要有常年累月的经验积累，职业壁垒比较高，对公司产生的价值往往是指数式的增长。
- 建议的学习路径：
 - 数据结构->leetcode刷题->算法应用入门《数学之美》->机器学习算法->大数据技术
 - 数据结构->leetcode刷题->大数据技术->算法应用入门《数学之美》->机器学习算法

3、算法相关岗位和面试小抄

- **算法相关岗位**（每个公司的岗位命名不同，看具体岗位要求）
 - 平台/数据处理类（数据工程师、数仓工程师、平台工程师、数据架构等）
 - 数据计算平台/引擎的搭建使用，数据的清洗、处理、统计，底层开发
JAVA/C++，大数据处理、分布式计算等
 - 算法研究与应用类（算法工程师/机器学习算法/深度学习算法/数据挖掘算法、推荐算法/图像算法/风控算法/广告算法/NLP、数据科学家/数据分析师）
 - 文本挖掘，自然语言处理、领域知识图谱、主题挖掘、社交网络分析、智能助手等
 - 图像处理，图像识别、音视频识别、无人驾驶等
 - 推荐，广告推荐、商品推荐、音乐推荐、新闻推荐、视频推荐
 - 排序，搜索排序、广告排序、商品排序等
 - 风控，信用评估、反欺诈、反作弊等
 - 商业智能，统计报表、数据趋势预测分析等
 - 传统行业应用，offline to online, online to offline, 用户画像，客户意愿预测等
- **面试小抄**
 - 简历准备（真实，不说废话，照着JD改的简历就是最好的简历）
 - 面试知识：

- 代码算法（基本算法/基本计算机知识/机器学习算法）
 - 功夫全在平时刷题和看一些技巧（《编程之美》《代码之美》《编程珠玑》《剑指offer》）
 - 遇到没见过的题怎么办？
 - 确定问题的边界、异常处理、测试用例先行
 - 先暴力解决（done is better than perfect）
 - 再考虑优化（时间复杂度优化，空间复杂度优化，使用特殊的数据结构 Hash、堆、栈、树）
 - 也可以问面试官提示
- 项目描述（STAR面试法，过去的行为是未来行为的最好预言）
 - Situation（背景）
 - Task（任务）
 - Action（行动）
 - 解决方案
 - 深入了解过程
 - 用了多久解决问题
 - 是否引入了新的技术方案
 - 遇到了哪些困难，如何克服的
 - 还有没有优化的空间
 - Result（结果）
- 业务理解和思考深度
- 沟通和表达能力
- 开放性问题/场景化问题解决（平时积累方法论）
- 面试官一般喜欢聪明、谦虚、不放弃的候选人
- 保持谦虚，永远高潜（不断空杯，勇于踏出舒适区接受挑战，拥抱变化，主动变化）
- 选择比努力更重要（谈offer谈意愿很重要）
 - 先挑行业，再挑公司，看准市场大环境（O2O、直播、P2P、区块链）
 - 每家公司都有或多或少的问题，即使光鲜的大公司可能也不尽如人意，快速适应公司文化才是生存之道
 - 换公司不能解决个人成长的问题，依然需要自我痛苦的蜕变（和领导不和、和同事不睦、做的事情不喜欢、接受不了公司文化、为什么升职加薪的不是我，凡事先从自我反思的人走得会更长远 路会更开阔）
 - 考虑offer，薪酬很重要，但不是最重要的，是不是自己喜欢做的感兴趣的事情能坚持下去的事情更重要，发展空间也非常重要
- 提升自己的影响力（Open and Connected，人品守恒定律）
 - 通过帮助别人
 - 通过各种场合的分享
 - Facebook五大核心价值观，工程师价值观：
 - Focus on Impact（做价值最大的事情）
 - Move Fast（快速落地，Done is better than perfect）
 - Be Bold（敢于冒险）
 - Be Open（保持开放）
 - Build Social Value（打造社交价值，Open and Connected）

4、答疑

- a. 解题思路&递归思路
 - “递归很难，简单的题也做不出来”

- “698. 划分为k个相等的子集” (<https://leetcode-cn.com/problems/partition-to-k-equal-sum-subsets/>)
 - 给定一个整数数组 nums 和一个正整数 k，找出是否有可能把这个数组分成 k 个非空子集，其总和都相等。返回True/False。
 - 递归方法：
 - 边界，判断数组和是否能整除k，不能的话直接返回false
 - 可以的话设置mark记录已经访问过的数字，要找子集的和subSum，当前子集的和curSum
 - 调用递归函数，如果k=1，说明数组直接是要找的子集，返回true
 - 当curSum==subSum时，说明已经找到了一个子集，将k-1继续传入递归函数
 - 退出，k=1时退出，或者mark标记中所有数字都访问过了
 - 什么问题可以用递归解决：
 - 子问题与原问题一样解决，且规模更小
 - 有程序停止条件（不能是死循环）
 - 递归要点：
 - 抽象重复的逻辑
 - 控制逻辑的边界
 - 恰当的退出
- “687. 最长同值路径” (<https://leetcode-cn.com/problems/longest-univalue-path/>)
 - 给定一个二叉树，找到最长的路径，这个路径中的每个节点具有相同值。这条路径可以经过也可以不经过根节点。返回边数。
 - 递归方法：
 - 抽象重复逻辑：从根节点开始，分别递归求出左右子树的最长同值路径，然后判断根节点和左右节点的关系，如果相等则加1，不相等就置为0，每次算完新的left和right后，要更新当前最大同值路径的值。注意返回的时候要返回左右中最大的数。
 - 控制逻辑的边界：
 - 坑一：如果当前节点与左节点的值不同，那么左边的同值路径就会断掉，同理，如果和右节点值不同，右边的同值路径就会断掉。一旦断掉，那么当前节点的同值路径值就应该清零。
 - 坑二：如果当前节点和父节点的值相同，需要注意，会从本节点向父节点连一个新的同值路径，与从子节点来的同值路径无关，不可累加。本题求解的是多条同值路径中最长的那条。
 - 坑三：同值路径必须是单向的，就是用纸笔能一笔画完的，不能倒退着连。
 - 恰当的退出：遍历完二叉树，到叶子节点为止。
- “再次，关于快排！”
 - 指定数组起始、结束和哨兵pivot
 - 根据哨兵将数组分成两部分，小于哨兵值的部分和大于哨兵值的部分，递归执行
 - 直到传入递归函数中数组的start==end结束
 - 初学者尽量不要写while套while，逻辑容易出错

b. 职业规划与学习

- “目前从事的是php开发，计划转岗去做算法工程师，那么需要具备什么样的能力，可以从哪些方面入手进行能力提升，面试官在考察的时候会注重哪些？”（通用的之前都讲过了，每个人的背景情况不同，可以再细聊下）
- “如何充分利用碎片化的时间来刷题、提高自己算法能力”（朋友圈打卡是个很好的方式）

c. 关于语言

- “用 C++ 做算法题用到的知识语法和内置或者STL里的数据结构”（有一个c++版的leetcode题解，语言知识也可以参考下 <https://github.com/soulmachine/leetcode>）
- 如果未来走算法研究应用路线的同学，在语言不受限的环境下，最好学习下python，用于做一些简单快速的算法实现和数据分析