

国内BAT、TMD等大厂在2023–2024年的算法面试中，普遍考察数组、字符串、链表、树、图、哈希表、栈、队列、堆等数据结构，以及排序、搜索、动态规划、贪心、回溯、双指针、滑动窗口、二分查找、分治、并查集、位运算等核心算法。题目难度主要集中在LeetCode的简单和中等水平，其中中等难度题目占比最高，旨在全面评估候选人的算法基础、编码能力、问题分析及优化思维。

## 国内大厂（BAT、TMD）算法面试真题分析（2023–2024）：类型与难度

### 1. 算法面试题概览与趋势

#### 1.1 大厂算法面试的普遍特点

国内大型互联网公司（通常指BAT：百度、阿里巴巴、腾讯；以及TMD：美团、滴滴、字节跳动等）的算法面试普遍具有较高的难度和广度，旨在全面考察候选人的计算机科学基础知识、问题解决能力、编码实现能力以及算法思维。这些面试通常不仅仅局限于对特定算法或数据结构的记忆，更强调在实际问题中灵活运用和优化算法的能力。根据多个信息来源，大厂算法面试题目的一个显著特点是其与LeetCode等在线编程平台上的题目难度相当，甚至有时会直接采用或改编自这些平台上的题目。面试过程中，候选人通常需要在有限的时间内（例如30分钟左右）手撕代码，理解问题、分析问题、设计算法、编写代码，并能解释其时间复杂度和空间复杂度。此外，面试官可能会针对候选人的解答进行追问，考察其对算法细节的理解深度以及优化能力。例如，在解决一个问题后，可能会被要求考虑更优的解法，或者在特定约束条件下调整算法。一些面试经验分享指出，字节跳动的面试中，算法题的难度通常较大，并且面试流程允许候选人在一个部门面试未通过后，快速转投其他部门，甚至有些部门无需笔试即可进入面试环节，这反映了算法能力在字节跳动等公司招聘中的核心地位。面试题库的更新频率相对较低，一些高频题目会被反复考察，这意味着针对性地准备高频题和经典题型是提升面试通过率的有效策略。

#### 1.2 2023–2024年算法面试趋势：从手撕代码到思维能力考察

近年来，特别是2023年至2024年期间，国内大厂的算法面试趋势逐渐从单纯的“手撕代码”向更全面的“思维能力考察”转变。虽然编码能力依然是基础，但面试官更加注重候选人在解题过程中展现的分析问题、设计解决方案以及优化算法的综合能力。一篇关于2024届一线互联网大厂校招算法题侧重点的文章明确指出，面试考察的重点正在从直接编写代码转向对思维能力的评估。这意味着，即使候选人能够快速写出代码，如果缺乏清晰的解题思路、对算法复杂度的准确分析，或者无法针对不同情况提出优化方案，也可能难以通过面试。例如，在技术二轮面试中，面试官不仅要求候选人完成一道难度较低的算法题，还会立即追问在看到题目时的思考过程，以及如何一步步地构建代码。这种趋势要求候选人在准备面试时，不仅要熟练各种算法和数据结构的实现，更要注重培养自己的逻辑思维和问题解决能力，能够清晰地阐述

自己的解题思路，并对算法的性能进行深入分析。此外，一些面试题会结合实际应用场景，考察候选人在特定业务背景下设计和优化算法的能力，例如滴滴面试中关于路径规划算法及其应用场景的问题，或者设计满足特定性能要求的缓存机制（如LRU算法）等。算法优化思维也备受关注，时间复杂度和空间复杂度的优化是重要的考察方向，尤其是在内存敏感的大数据场景下，如何在时间与空间之间进行权衡是面试官关注的重点。

## 2. BAT（百度、阿里巴巴、腾讯）算法面试真题分析

### 2.1 百度（Baidu）

#### 2.1.1 常见算法题类型与示例

百度在2023年至2024年的算法面试中，考察的题目类型比较广泛，涵盖了数据结构、算法设计以及特定场景应用等多个方面。根据2023年6月的一篇博客园文章，百度在其面试中会考察**限流算法**，例如漏桶算法、令牌桶算法和计数器算法。这篇文章不仅提及了这些算法的概念，还提供了Java代码实现示例。例如，**计数器限流算法**的实现中，通过维护一个起始时间和一个时间间隔内的请求数量，来判断是否超出限制。当新的请求到达时，如果当前时间在时间间隔内且请求数未达上限，则允许通过；否则，判断为超出限制。**令牌桶限流算法**则维护一个固定容量的桶，以恒定速率向桶中添加令牌。请求到达时，如果桶中有足够令牌则允许通过并消耗相应数量的令牌；否则，请求被限流。**漏桶算法**则将请求视为水滴，以固定速率从桶中漏出。如果桶未满，则请求可以进入；否则，请求被限流。这些算法都属于系统设计和并发控制领域，考察候选人对高并发场景下流量控制的理解和实现能力。此外，根据2023年算法岗社招面经分享，百度在面试中也会考察经典的LeetCode题目，如“3sum”问题，主要考察对数组的处理和双指针技巧的运用。另一道被提及的题目是“岛屿最大面积”，这是一道常见的图遍历问题，通常使用深度优先搜索（DFS）或广度优先搜索（BFS）来解决。链表相关的问题出现频率也较高，例如“反转链表”这道题目，既可能要求迭代实现，也可能要求递归实现。动态规划也是百度的常考知识点，例如“编辑距离”问题，面试官会关注候选人对状态定义、状态转移方程的理解，以及能否清晰地解释DP数组的含义和更新过程。字符串处理相关的题目也有所涉及，例如判断一个字符串能否通过重新排列组成另一个特定字符串（如“Baidu”）。

#### 2.1.2 面试题难度分析

从提供的限流算法示例来看，百度面试中涉及的算法题目具有一定的实践性和深度。虽然限流算法本身的概念相对清晰，但面试官可能会深入考察候选人对这些算法细节的理解，例如不同算法的优缺点、适用场景、以及在实际编码中可能遇到的问题（如并发安全、精度控制等）。文章中提供的Java代码实现涉及了多线程模拟测试，这表明面试中可能会要求候选人具备将算法思想转化为实际可运行代码的能力，并且需要考虑并发环境下的正确性。虽然这些限流算法在LeetCode等平台上可能不常见，但其背后的思想（如计数、速率控制、队列管

理) 与许多经典算法问题有共通之处。因此, 可以推断百度算法面试的难度不仅在于对经典算法的掌握, 还在于对特定领域 (如系统设计、并发编程) 相关算法的理解和应用。一篇CSDN博客提及了2024年百度、头条、小米、360、网易等公司的Android面试题, 但未详细展开算法部分, 仅提及了“算法策略”, 这暗示了算法在面试中的重要性。从2023年算法岗社招面经来看, 百度的算法面试题难度整体较高, 需要候选人进行充分的准备, 面经中明确提到“**百度的代码题还是最需要好好准备的**”。虽然提及的“3sum”和“岛屿最大面积”在LeetCode上通常被归类为中等难度, 但考虑到百度还会考察一些“没见过”的题目, 这无疑增加了面试的难度, 因为这些题目往往需要候选人在有限的时间内进行独立思考和创新性解决。此外, 面试中除了算法题, 还会涉及机器学习项目、大模型应用等领域的讨论, 这也要求候选人具备广泛的技术知识和项目经验, 进一步提升了整体的面试难度。

## 2.2 阿里巴巴 (Alibaba)

### 2.2.1 常见算法题类型与示例

阿里巴巴的算法面试题覆盖了从基础数据结构到高级机器学习算法的广泛范围。一篇2024年3月的CSDN博客详细解析了阿里巴巴春招深度学习的面试题, 涉及深度学习基础、模型架构理解、优化算法与训练技巧、正则化技术等。例如, 在正则化技术方面, 面试题要求解释L1、L2正则化和Dropout在防止过拟合中的作用, 并比较它们的区别。L1正则化倾向于产生稀疏权重, 有助于特征选择; L2正则化使权重接近于0但不完全为0, 有助于减少模型复杂度; Dropout则通过随机丢弃神经元来增强模型的泛化能力。另一篇2024年5月的CSDN博客则包含了一些更偏向于传统算法和数据结构的题目, 例如, 给定**4亿个int数, 如何找出重复的数** (可以使用hash方法, 建立一个2的32次方个bit的hash数组)。还有题目涉及**从1万个降序数组中找出最大的N个数** ( $N < 1000$ )。此外, 阿里云的开发者社区也分享了2022年2023届秋招的算法笔试题, 例如LeetCode 2038题 (如果相邻两个颜色均相同则删除当前颜色)。根据2023年算法岗社招面经, 阿里巴巴 (具体为阿里BU1和BU2) 在算法面试中, 除了对项目和特定技术领域 (如机器学习、强化学习) 进行深入考察外, 也会涉及算法编程题。常见的算法题类型包括**树的遍历** (例如蚂蚁集团面试中出现的“树的某种遍历”) 和**动态规划** (例如阿里BU1面试中考察的“mid难度动态规划”题目)。牛客网上的信息也显示, 阿里巴巴的编程题难度从2星到4星不等, 涉及的知识点可能包括但不限于数组、字符串、链表、树、图、动态规划、搜索、贪心算法等。

### 2.2.2 面试题难度分析

阿里巴巴的算法面试题难度跨度较大, 既有考察基础编程能力的题目, 也有需要深入理解算法原理和具备较强问题解决能力的题目。例如, 找出4亿个int数中重复的数, 如果采用哈希方法, 需要考虑内存限制, 并巧妙地利用位图 (bitmap) 来减少空间占用, 这考察了候选人对数据特性和算法应用场景的理解。而从1万个降序数组中找出最大的N个数, 则需要有效利用各个数组的有序性, 可能涉及到堆 (优先队列) 等数据结构的运用, 对算法的设计和优化能

力有一定要求。在深度学习领域，面试题不仅要求解释概念，如L1、L2正则化和Dropout的原理，还可能涉及到这些技术在实际模型中的应用和效果分析，这需要候选人对相关理论有扎实的掌握。LeetCode 2038这类题目虽然来自题库，但也需要候选人能够快速理解题意并给出正确的解法，属于中等难度。总体而言，阿里巴巴的算法面试注重候选人对算法和数据结构的扎实掌握，以及在具体问题中灵活运用和优化算法的能力，同时也会根据岗位需求考察特定领域的专业知识。阿里巴巴的算法面试题难度整体偏高，尤其对于P7及以上级别的岗位，面试官会期望候选人对算法有深入的理解和丰富的实践经验。即使是常规的数据结构和算法题目，也可能通过增加限制条件、要求多种解法、或者结合具体业务场景来提高难度。从2023年算法岗社招面经来看，阿里BU1明确提到了“**mid难度动态规划**”，这表明其算法题目的难度至少是LeetCode中等水平。蚂蚁集团的“树的某种遍历”虽然属于基础题型，但在实际面试中，面试官可能会通过追问优化方式、边界条件处理、非递归实现等来提升问题的综合难度。阿里巴巴的面试非常注重项目经验的挖掘，算法题往往只是考察的一部分，面试官会更看重候选人解决实际问题的能力以及在项目中应用算法的经验。

## 2.3 腾讯（Tencent）

### 2.3.1 常见算法题类型与示例

腾讯的算法面试题同样涵盖了广泛的数据结构和算法知识。一篇2023年11月发布的牛客网帖子整理了腾讯后台2024秋招的面试题，其中算法题包括**几何计算和字符串处理**。例如，“一根弦切割圆两部分，给出两部分弓形面积比例，求弦长”考察了数学几何知识和计算能力。“字符串s包括字母和数字，可以将字母换成大小写，给出了全部方案”则是一道典型的回溯算法或位运算相关的题目，要求生成所有可能的组合。另一篇CSDN博客汇总了2024年腾讯算法岗的面试题，其中非技术问题中也包含了一些逻辑思维题，如“5枚硬币，2正3反如何划分为两堆然后通过翻转让两堆中正面向上的硬币和反面向上的硬币个数相同”以及“N\*N的方格纸,里面有多少个正方形”，这些问题虽然不是传统意义上的编程算法题，但也考察了候选人的分析问题和逻辑推理能力。此外，一篇更早的关于腾讯算法面经的文章（2017年）提到了大量传统机器学习算法，如随机森林（RF）、XGBoost、逻辑回归（LR）等，并深入探讨了它们的原理、区别、优缺点和适用场景，例如RF和XGBoost的区别，XGBoost如何给出特征重要性等。CSDN博客和牛客网上的信息也显示，腾讯的算法面试题可能涉及字符串处理（如“**最小覆盖子串**”）、链表操作（如“**反转链表**”、“**删除链表的倒数第N个结点**”）、二叉树（如“**二叉树的最大深度**”、“**路径总和 II**”）、动态规划（如“**斐波那契数**”、“**最大子数组和**”、“**打家劫舍**”等基础题型）以及一些实际应用场景的抽象，例如“**给阿里2万多名员工按年龄排序应该选择哪个算法？**”。

### 2.3.2 面试题难度分析

腾讯算法面试题的难度因岗位和具体题目而异。例如，几何计算题“一根弦切割圆两部分，给出两部分弓形面积比例，求弦长”，如果候选人没有相关的数学知识背景或者不能将其转化为

数学模型，可能会觉得有一定难度。字符串大小写转换并生成所有方案的题目，如果采用回溯算法，需要清晰地定义状态和选择，并正确实现递归和回溯过程，属于中等难度的题目。而逻辑思维题如“5枚硬币”问题，则更考验候选人的灵感和巧思，可能对于不熟悉这类问题的候选人来说有一定挑战。在机器学习算法方面，深入探讨RF和XGBoost的区别、XGBoost的特征重要性计算方式等问题，则要求候选人对这些算法的内部原理有深刻的理解，不仅仅是调包侠，这属于较高难度的考察。一篇腾讯云社区的文章提到了BAT算法面试题中的环形链表问题，并给出了哈希表法和不使用额外空间的快慢指针法，这属于链表操作中的经典问题，难度中等，但快慢指针法对思维要求较高。总体来看，腾讯的算法面试既包含对基础算法和数据结构的考察，也包含对逻辑思维和特定领域（如机器学习）算法原理的深入追问，难度中等偏上。例如，“最小覆盖子串”这类滑动窗口问题，虽然思路清晰，但在实现过程中需要注意很多细节，如字符计数的更新、窗口的移动条件等，对编码能力有一定要求。二叉树和动态规划的题目，如果涉及到一些变种或者优化，难度也会相应增加。对于一些结合数学知识的题目，如求弦长的问题，需要候选人具备一定的数学推导能力和将数学问题转化为编程问题的能力。

### 3. TMD（美团、滴滴、字节跳动）及其他大厂算法面试真题分析

#### 3.1 美团（Meituan）

##### 3.1.1 常见算法题类型与示例

美团的算法面试题以其注重实际应用和一定的难度而闻名。根据2024年的面试情况，美团在算法面试中倾向于考察候选人的**动态规划建模能力、空间复杂度优化以及数组遍历技巧**。例如，“**不同路径**”（LeetCode 63）就是一个典型的动态规划题目，考察候选人如何构建DP表来记录到达每个位置的路径数。另一个常见的题目是“**接雨水**”（LeetCode 42），这道题不仅考察对数组的操作，更侧重于如何在优化空间复杂度的前提下解决问题，通常会引导候选人思考如何使用双指针或栈结构来减少额外空间的使用。此外，“**字符串相加**”（LeetCode 415）也是美团面试中可能出现的问题，主要考察候选人对大数处理和进位机制的掌握程度，解题思路通常是从尾到头逐位计算并妥善处理进位。一篇2024年3月的CSDN博客详细解析了美团2024年春招第一场笔试的算法题，题目包括“**小美的平衡矩阵**”、“**小美的数组询问**”、“**小美的MT**”、“**小美的朋友关系**”以及“**小美的区间删除**”。这些题目通常以实际业务场景为背景，例如“**小美的区间删除**”要求计算删除某个区间后，剩余数字乘积末尾0的个数，这涉及到质因数分解（特别是2和5的个数）、前缀和以及双指针等算法思想。另一篇2023年3月的稀土掘金文章则分享了美团2024届暑期实习后端笔试的题目，例如“**小美观测流星**”问题，需要将流星的出现和消失时间转换为事件，然后进行排序和遍历统计，考察了事件处理、排序和贪心算法的思想。此外，一篇2024年4月的CSDN博客回忆了美团Android岗的面试，其中手写算法题是随机抽取的，一道示例题目是：“给出一个序列包含n个正整数的序列A，然后给出一个正整数x，你可以对序列进行任意次操作的，每次操作你可以选择序列中的一个数字，让

其与x做按位或运算。你的目的是让这个序列中的众数出现的次数最多。请问众数最多出现多少次？””。这道题考察了位运算以及通过操作改变数组元素以优化特定目标的能力。根据牛客网的讨论，美团的笔试和面试中还会出现图论相关的题目，例如“**小美的朋友关系**”，这类问题通常需要运用\*\*并查集（Union-Find）\*\*等数据结构来处理节点间的连接关系。

### 3.1.2 面试题难度分析

美团的算法题目整体难度被评价为“**没有想象中那么难**”，大约等同于中学信息学竞赛（OI）的普及组难度。然而，这并不意味着题目简单，一些题目仍然需要巧妙的算法思想和细致的编码实现。例如，“**小美的区间删除**”问题，虽然解题思路（统计2和5的个数，使用前缀和和双指针）相对明确，但在实现细节上，如如何高效计算乘积末尾0的个数，以及如何利用容斥定理和数论知识进行优化，对候选人的综合能力有一定要求。该问题的作者指出其最坏时间复杂度为 $O(w \cdot n)$ ，其中 $w = \log(1e9)$ ，这表明面试官可能会关注算法的时间复杂度。“**小美观测流星**”问题则需要将实际问题抽象为算法模型，并采用合适的排序和遍历策略，其时间复杂度为 $O(n \log n)$ 。而Android岗面试中的“**按位或运算求众数**”问题，则需要候选人理解位运算的特性，并分析如何通过操作使得某个数成为众数，具有一定的思维难度。有面试者反馈，美团的手写算法在业界算是比较出名的，很多候选人可能卡在手写算法上。这表明美团对候选人的实际编码能力和算法应用能力有较高要求。从美团在2024年考察的算法题来看，其难度主要集中在**中等水平**，侧重于考察候选人对基础算法的理解和应用能力，以及在特定场景下对算法进行优化的思维。例如，“**不同路径**”（LeetCode 63）和“**字符串相加**”（LeetCode 415）在LeetCode上均被标记为中等难度，而“**接雨水**”（LeetCode 42）则被标记为困难，但通常面试中可能会简化问题或给予更多提示，使其实际考察难度更偏向中等。这些题目要求候选人不仅能够给出解决方案，还需要分析算法效率，并思考优化方案。

## 3.2 滴滴（Didi）

### 3.2.1 常见算法题类型与示例

滴滴的算法面试题涵盖了数据结构、算法思想以及一些智力题。一篇2017年的面经秘籍（虽然年份较早，但部分内容仍有参考价值）列举了多种类型的题目。在线性表方面，涉及链表操作如**反转链表**、**单链表排序（要求 $O(n \log n)$ 时间复杂度）**、**合并两个链表**，以及将奇数位升序、偶数位降序的链表重排成升序链表。字符串处理方面，有求包含不超过k个不同字符的最长子串长度（LeetCode上的中等或偏难题目，可使用滑动窗口思路）、求和大于k的子串的最小长度、最长公共子串、以及括号匹配等问题。算法思想实战方面，提到了一个亿级节点按分数分档的问题、背包问题变体、数三减一（约瑟夫环问题）、国王挖金矿（动态规划）、最长递增子序列、绳子剪裁求最大乘积等。智力题方面，有铺砖问题（动态规划求解）、毒药找猪问题等。此外，一篇2023年12月的博客园文章提到了滴滴面试中关于**路径规划算法及其应用场景**的问题，这直接关联到滴滴的核心业务。根据2023年至2024年的面试经验分享，滴滴在算法面试中考察的题目类型多样，涵盖了链表、字符串、动态规划等常见算法领域。

例如，一道常见的题目是“**合并有序节点链表**”，这主要考察候选人对链表数据结构的理解和操作能力，以及合并算法的实现。另一道高频题目是“**最长回文子串**”，例如给定字符串 "aabccbabb"，要求找出最长的回文子串，这道题通常考察动态规划或中心扩展等算法思想。此外，滴滴的面试中也可能出现与特定业务场景结合的SQL题目，例如设计数据库存储结构以满足特定查询需求，或者在已有数据表（如订单表包含uid, oid, price）上进行查询，如找出特定用户的最大消费金额或订单数量最多的用户。

### 3.2.2 面试题难度分析

滴滴的算法面试题难度呈现多样化，既有经典的LeetCode风格题目，也有结合业务场景的实际问题，以及考察思维灵活性的智力题。链表和字符串相关的题目，如反转链表、单链表排序、滑动窗口求最长子串等，属于面试中常见的题型，难度一般为中等，要求候选人熟练掌握相关数据结构的操作和算法思想。动态规划类问题，如国王挖金矿、最长递增子序列、铺砖问题等，对候选人的问题建模能力和状态转移方程推导能力有较高要求，难度通常在中到高。智力题如毒药找猪问题，则更侧重于逻辑推理和问题分析能力，可能没有固定的算法模板，需要候选人临场发挥。关于路径规划算法的问题，由于直接关联到滴滴的核心业务，面试官可能会期望候选人对常见的路径规划算法（如Dijkstra、A\*等）有所了解，并能分析其在不同场景下的适用性，这属于有一定专业背景要求的题目。总体而言，滴滴的算法面试既考察基础算法和数据结构的掌握程度，也注重候选人的问题解决能力和逻辑思维能力，部分题目可能具有一定的挑战性。滴滴的算法面试题难度整体上属于**中等水平**，部分题目可能涉及对基础算法的灵活运用和一定程度的思维难度。例如，“**合并有序节点链表**”属于链表操作中的经典问题，难度通常被认为是简单到中等。而“**最长回文子串**”问题（如LeetCode 5）在LeetCode上被标记为中等难度，其动态规划解法需要一定的技巧性。滴滴面试中出现的SQL题目，如果涉及到数据库设计优化，则可能具有一定的挑战性。

## 3.3 字节跳动 (ByteDance)

### 3.3.1 常见算法题类型与示例

字节跳动以其算法面试难度较高而闻名，考察范围广泛且深入。根据2024年字节跳动的部分面试题目分析，其算法面试题主要考察候选人对基础数据结构的理解和编码能力，以及对常见算法的掌握程度和应用能力。常见的题目类型包括但不限于：**数组与矩阵操作**（如“**搜索二维矩阵**” LeetCode 74）、**二叉树相关问题**（如“**判断二叉树的值相等**” LeetCode 100；“**求根节点到叶节点数字之和**” LeetCode 129）、**基础数据结构概念理解**（如“**数组和链表的区别**”）、**树的遍历与路径总和**。此外，根据一份来自稀土掘金的分析，其考察重点可以归纳为**哈希表与字符串**（如“**无重复字符的最长子串**” LeetCode 3；“**最小窗口子串**” LeetCode 76）、**搜索算法 (DFS/BFS)**（如“**岛屿数量**” LeetCode 200）、**动态规划**（如“**爬楼梯**” LeetCode 70；“**打家劫舍**” LeetCode 198）、**复杂数据结构**（如并查集、线段树、Trie树）、**位运算、数组操作、智力题以及数学相关题目**。2024年字节跳动的面试真题中还出现

了“重排链表”，这道题目要求在不改变节点内部值的情况下，将链表  $L_0 \rightarrow L_1 \rightarrow \dots \rightarrow L_{n-1} \rightarrow L_n$  重新排列为  $L_0 \rightarrow L_n \rightarrow L_1 \rightarrow L_{n-1} \rightarrow L_2 \rightarrow L_{n-2} \rightarrow \dots$  的形式。在更高级的面试轮次中，可能会涉及到更复杂的问题，例如在  $M \times N$  的横向纵向均递增的矩阵中查找指定数，或者设计支持大规模评论系统的后端服务，包括数据库表设计、索引设计、分库分表策略以及深分页实现等。

### 3.3.2 面试题难度分析

字节跳动的算法面试题被普遍认为是**难度较大的**。即使是基础的数据结构题目，如计算二叉树的最大深度、岛屿的最大面积，也可能在面试的紧张环境下对候选人的熟练度和准确性提出考验。动态规划类题目，如最大子数组和、最长回文子串，是LeetCode上的经典题目，通常需要候选人掌握状态定义和状态转移方程的构建，难度在中到高。实现 lodash.get 方法则考察了候选人对JavaScript对象属性的访问和异常处理能力，属于偏应用型的题目。在深度学习领域，要求手写卷积代码、前向传播和反向传播代码，以及推导BN (Batch Normalization) 的内部实现等，对候选人的理论基础和动手能力要求极高，属于高难度范畴。LeetCode 74 (搜索二维矩阵) 和LeetCode 129 (求根节点到叶节点数字之和) 这类题目，虽然可能在题库中出现过，但在面试中需要在有限时间内清晰、准确地完成，并分析复杂度，也具有一定的挑战性。有面试经验分享指出，字节跳动的面试算法题**难度较大**，并且面试流程允许反复打捞，也从一个侧面反映了其算法考核的严格性。根据分析，面试题目偏向于**简单和中等难度**，难题 (Hard) 占比较低，约为11.7%。这是因为面试时间通常有限，过于复杂的题目难以在短时间内完成。然而，这并不意味着面试简单，面试官更看重解题过程的清晰度、代码的鲁棒性以及对时间空间复杂度的精确分析。面试官可能会对题目进行改编，或者要求候选人给出多种解法并进行比较。此外，面试题库更新频率相对较低，一些高频经典题型会被反复使用。

## 3.4 其他大厂（如小红书）算法面试题特点

除了BAT和TMD这些头部大厂，其他快速发展的互联网公司如小红书，在算法面试方面也形成了自己的特点。根据知乎专栏的一篇文章，小红书2024年的算法面试题同样注重对基础算法和数据结构的考察，并结合了实际应用场景。例如，**会议室的安排问题** (LeetCode 763的变体，通常指类似LeetCode 253. 会议室 II 或LeetCode 452. 用最少量的箭引爆气球的问题)，考察了事件调度和贪心算法的应用，要求候选人能够将实际问题抽象为算法模型，并给出有效的解决方案。在**排序数组中查找元素的第一个出现位置** (LeetCode 35) 以及**找到排序好的数组中第一个大于设定目标值的下标位置** (自定义问题)，这两道题都考察了二分查找算法的实现及其变体，对候选人的编码细节和边界条件处理能力有较高要求。这些题目表明，小红书的算法面试倾向于考察候选人对经典算法的掌握程度，以及在实际问题中应用这些算法的能力。虽然具体的题目难度可能因岗位和面试轮次而异，但整体上可以看出其对算法基础的重视。这类公司通常也面临着快速迭代和业务扩张的需求，因此需要员工具备扎实的技术功底和快速解决问题的能力，算法面试正是评估这些能力的重要手段。小红书的算法面试题在

难度上可能不是极端困难，但非常注重对基础算法的扎实掌握和灵活运用，以及在编码过程中对细节和边界条件的把控。

## 4. 算法面试题主要类型与难度分布

### 4.1 数据结构类题目

数据结构是算法面试的基石，国内大厂面试中对此类题目的考察非常普遍。候选人需要熟练掌握各种常用数据结构的特性、基本操作及其应用场景。

#### 4.1.1 数组与字符串

数组和字符串是算法面试中最常见的数据结构。题目类型多样，包括但不限于：查找（如二分查找、搜索旋转排序数组）、排序（如归并排序、快速排序的应用）、子数组/子串问题（如最大子数组和、最长无重复字符子串、最小覆盖子串）、双指针技巧（如两数之和、三数之和、移除元素）、滑动窗口（如找到字符串中所有字母异位词）、以及各种模拟和操作题。例如，腾讯面试中出现过“字符串s包括字母和数字，可以将字母换成大小写，给出了全部方案”，这需要候选人能够有效地遍历和操作字符串，并生成所有可能的组合。字节跳动面试中出现的“实现 lodash.get 方法”也间接考察了对数组和对象（可视为键值对集合，与字符串操作有共通之处）的访问和处理能力。美团笔试中的“小美的数组询问”和“小美的区间删除”都直接以数组为操作对象，考察了前缀和、双指针等技巧。在字节跳动的面试中，与数组和字符串相关的题目非常普遍，如“无重复字符的最长子串”（LeetCode 3），“最小窗口子串”（LeetCode 76）等。此外，剑指Offer中也包含大量数组和字符串相关的题目，如“二维数组的查找”（剑指Offer 1），“替换空格”（剑指Offer 2），“顺时针打印矩阵”（剑指Offer 19）等。根据知乎专栏文章《刷完这些 Leetcode 题，FAANG TMD BAT 算法必过》的分类，数组相关的题目可以细分为旋转、前缀和、统计元素、改变和移动、遍历、二维数组及滚动数组、特定顺序遍历二维数组、二维数组变换等多个子类别。字符串相关的题目则包括字符处理、回文串判断、公共前缀、单词处理、字符串反转、子序列判断以及高精度运算等。

#### 4.1.2 链表

链表相关的问题也是大厂面试的热点，尤其喜欢考察指针操作和边界条件处理。常见题目包括：**反转链表**（部分或全部、迭代和递归两种方式）、**合并两个有序链表**、链表的中间节点、**环形链表的判断与入口查找**、相交链表的判断、**删除链表的倒数第N个节点**、以及各种复杂的链表排序（如归并排序）和重排问题。例如，BAT面试题中曾出现“环形链表”问题，并提示尝试不使用额外空间解决，这通常指向快慢指针法。滴滴的面经中也提到了反转链表、单链表排序（要求O(nlogn)）、合并链表以及将特定顺序的链表重排为升序等问题。这些问题不仅考察对链表结构的理解，还考察了候选人对指针操作的熟练度和代码实现的鲁棒性。CSDN博客中整理的BAT大厂面试题中，明确列出了“手写链表逆序代码”、“判断单链表成环与否？”、“链表翻转（即：翻转一个单项链表）”以及“合并多个单有序链表（假设都是

递增的）”。字节跳动2024年的面试真题中也出现了“重排链表”。根据知乎专栏文章《刷完这些 Leetcode 题，FAANG TMD BAT 算法必过》的分类，链表相关的题目可以细分为删除或插入、遍历、旋转与反转、加法、合并、双指针等多个子类别。

### 4.1.3 树与二叉树

树和二叉树是面试中非常重要的数据结构，相关题目往往涉及到递归和遍历。常见考点包括：二叉树的三种深度优先遍历（前序、中序、后序）及其迭代写法、广度优先遍历（层序遍历）、二叉树的最大深度/最小深度、判断二叉树是否对称/平衡、二叉树的最近公共祖先、路径总和问题（如求根到叶子节点数字之和）、二叉搜索树（BST）的验证、插入、删除、以及BST相关性质的应用（如第K小的元素）。例如，字节跳动面试中出现了“计算二叉树的最大深度”和“判断二叉树的值相等（LeetCode 100）”。这些题目要求候选人对树的结构和遍历算法有清晰的理解，并能熟练运用递归或迭代进行求解。CSDN博客整理的BAT大厂面试题中也包含了“二叉树给出根节点和目标节点，找出从根节点到目标节点的路径”、“二叉树的深度优先遍历和广度优先遍历的具体实现”、“讲一下对树，B+树的理解”。根据知乎专栏文章《刷完这些 Leetcode 题，FAANG TMD BAT 算法必过》的分类，树与二叉树相关的题目可以细分为遍历（前序、中序、后序、层序）、搜索、路径问题、修改或构造、以及二叉搜索树（BST）特性等多个子类别。

### 4.1.4 图

图相关的问题在大厂面试中也占有一定比例，尤其是在一些对图算法有特定需求的岗位（如地图、社交网络、推荐系统等）。常见题目包括：图的表示（邻接表、邻接矩阵）、图的深度优先搜索（DFS）和广度优先搜索（BFS）、拓扑排序、最短路径算法（Dijkstra、Bellman-Ford、Floyd-Warshall）、最小生成树算法（Prim、Kruskal）、以及一些经典的图论问题（如二分图判断、环路检测）。字节跳动面试中出现的“岛屿的最大面积”就是一个典型的图的遍历问题，可以使用DFS或BFS解决。虽然基础图算法的直接考察可能不如数组、字符串、链表和树那样频繁，但图的思想和遍历技巧在很多其他问题中也有应用。CSDN博客中整理的BAT大厂面试题中提到了“讲一下对图的理解”。根据知乎专栏文章《刷完这些 Leetcode 题，FAANG TMD BAT 算法必过》的分类，图相关的题目可以细分为图的表示、广度优先搜索（BFS）、深度优先搜索（DFS）、最小生成树、最短路径以及连通性等子类别。

### 4.1.5 栈、队列与堆

栈、队列和堆（优先队列）是三种非常重要的辅助数据结构，它们在很多算法中扮演着关键角色。栈常用于模拟递归、表达式求值、括号匹配等场景。队列则常用于BFS、滑动窗口最大值等问题。堆（优先队列）常用于求Top K问题、合并K个有序链表、Dijkstra算法等需要高效获取最大或最小元素的场景。例如，美团面试中“小美的朋友关系”问题，虽然原文未给出具体解法，但这类涉及关系处理和查询的问题，有时会用到并查集（一种特殊的树形结构，也可

视为一种集合的表示) 或基于堆的优化。虽然直接要求实现这些数据结构的题目不多, 但它们在解决更复杂问题时是不可或缺的工具。腾讯的面试题中曾出现过“滑动窗口最大值”, 这题可以使用单调队列或堆来高效解决。根据知乎专栏文章《刷完这些 Leetcode 题, FAANG TMD BAT 算法必过》的分类, 栈和队列相关的题目可以细分为前K个高频元素 (优先队列/堆) 、递归 (栈的应用) 、表达式求值、字符串去重、单调栈、滑动窗口最大值、用栈访问最后元素、栈与括号匹配、最大/最小栈以及实现栈/队列等子类别。

#### 4.1.6 哈希表

哈希表 (散列表) 以其高效的查找和插入性能, 在算法设计中应用广泛。面试中常考察哈希表的原理、冲突解决方法 (如链地址法、开放寻址法) 、以及如何设计一个好的哈希函数。题目类型包括: 两数之和、存在重复元素、最长连续序列、字母异位词分组 等。例如, 阿里巴巴面试题中“4亿个int数, 如何找出重复的数”, 一种高效的解法就是利用哈希思想, 通过位图 (bitmap) 实现。BAT面试题中环形链表的哈希表解法也是利用哈希表来检测重复节点。哈希表不仅是一种独立的数据结构, 更是一种重要的算法思想, 能够显著优化许多问题的求解效率。在字节跳动的面试中, 哈希表与字符串常常结合考察, 如“最长回文串” (LeetCode 409) 、“同字符词语分组” (LeetCode 49) 等题目都巧妙地运用了哈希表来降低时间复杂度。根据知乎专栏文章《刷完这些 Leetcode 题, FAANG TMD BAT 算法必过》的分类, 哈希表相关的题目可以细分为查找、插入、删除、索引、统计以及结合前缀和等子类别。

### 4.2 算法类题目

算法类题目是面试的核心, 考察候选人对不同算法思想的掌握和应用能力。

#### 4.2.1 排序与搜索

排序和搜索是算法的基础, 面试中常考。排序算法方面, 要求掌握常见排序算法 (如冒泡排序、选择排序、插入排序、希尔排序、归并排序、快速排序、堆排序、计数排序、桶排序、基数排序) 的原理、时间复杂度、空间复杂度、稳定性及其适用场景。搜索算法方面, 二分查找是重中之重, 不仅要求掌握标准的二分查找, 还要能处理其各种变体 (如查找第一个/最后一个等于目标值的元素、查找第一个大于/小于目标值的元素等)。例如, 字节跳动面试中出现的“搜索二维矩阵 (LeetCode 74) ” 和“在排序数组中查找元素的第一个出现位置 (LeetCode 35) ” 都考察了二分查找及其变体。一篇CSDN博客总结的2024年大厂面试必备知识中也包含了十大排序算法。在力扣的讨论中, 提到了枚举时留意单调性可以考虑二分查找。

#### 4.2.2 动态规划

动态规划 (DP) 是面试中的难点和重点, 考察候选人的问题分解、状态定义和状态转移方程构建能力。常见DP问题包括: 背包问题 (01背包、完全背包) 、最长递增子序列 (LIS) 、最

长公共子序列（LCS）、编辑距离、最大子数组和、股票买卖系列问题、打家劫舍系列问题、以及各种路径问题。例如，字节跳动面试中出现的“最大子数组和”和“最长回文子串”都是经典的动态规划问题。滴滴面经中也提到了“国王挖金矿的变体”和“最长递增子序列”。掌握动态规划需要大量的练习和总结，理解其“最优子结构”和“重叠子问题”的核心思想。CSDN博客文章《人人都在求学的2024年大厂面试必问算法...》中，第四章专门讲解了递归和动态规划，列举了大量经典DP问题，如“斐波那契数列问题的递归和动态规划”、“矩阵的最小路径和”、“换钱的最少货币数”、“最长递增子序列”、“最长公共子序列问题”、“最长公共子串问题”、“编辑距离”（对应“最小编辑代价”），“背包问题”（对应“换钱的方法数”）等。

#### 4.2.3 贪心算法

贪心算法通常用于求解最优化问题，它在每一步都做出在当前看来是最好的选择，希望最终得到全局最优解。贪心算法的难点在于证明其正确性。常见贪心问题包括：区间调度问题（如会议室安排）、分糖果、跳跃游戏、买卖股票的最佳时机II（可以多次交易）、Huffman编码等。美团面试中的“小美观测流星”问题，通过将流星的出现和消失时间转换为事件并进行排序处理，也体现了贪心算法的思想，即优先处理结束时间早的事件以最大化观测数量。贪心算法通常代码实现相对简单，但需要敏锐的洞察力来判断是否适用。CSDN博客文章《人人都在求学的2024年大厂面试必问算法...》中，动态规划章节中提及的“换钱的最少货币数”等问题，在某些特定条件下也可以考虑贪心解法。

#### 4.2.4 回溯算法

回溯算法是一种通过探索所有可能的候选解来找出所有解的算法。如果候选解被确认不是一个解（或者至少不是最后一个解），回溯算法会通过在上一步进行一些变化来丢弃该解，即“回溯”并尝试其他可能的选项。回溯算法通常与递归结合使用，是解决组合问题、排列问题、子集问题、切割问题、棋盘问题（如N皇后问题）等的常用方法。例如，一位后端面试者总结中提到的“复原IP地址”就是一个经典的回溯算法问题。根据知乎专栏文章《刷完这些Leetcode 题，FAANG TMD BAT 算法必过》的分类，回溯算法通常与递归放在一起讨论。例如，在递归部分列举的回溯问题有：“51. N 皇后”（60分），“39. 组合总和”（80分），“46. 全排列”（80分）。CSDN博客文章《人人都在求学的2024年大厂面试必问算法...》中，第四章“递归和动态规划”部分也包含了大量可以用回溯法解决的问题，如“N皇后问题”。

#### 4.2.5 双指针与滑动窗口

双指针（Two Pointers）和滑动窗口（Sliding Window）是两种常用的处理数组或链表问题的技巧。双指针技巧通常用于在有序数组中查找满足特定条件的元素对（如两数之和、三数之和）、判断链表是否有环、找到链表的中间节点等。滑动窗口技巧则常用于解决子数组或子串相关的问题，如找到满足条件的最短/最长子数组（如“长度最小的子数组”、“最小覆盖子串”）、字符串的排列或异位词查找等。这两种技巧都能有效地将时间复杂度从 $O(n^2)$ 降低

到 $O(n)$ 或 $O(n\log n)$ 。例如，腾讯面试中出现的“三数之和”和“最小覆盖子串”都是典型的应用场景。力扣的讨论中也提到，双重循环可以看看是否可以用双指针或者前缀和之类的算法优化。根据知乎专栏文章《刷完这些 Leetcode 题，FAANG TMD BAT 算法必过》的分类，双指针可以细分为对撞指针和快慢指针。滑动窗口则分为固定窗口大小和可变窗口大小两种。

#### 4.2.6 二分查找

二分查找（Binary Search）是一种在有序数组中查找特定元素的高效算法。其基本思想是将目标值与数组中间元素进行比较，如果相等则返回，如果目标值小于中间元素，则在左半部分继续查找，否则在右半部分继续查找，直到找到目标值或搜索区间为空。大厂面试中，二分查找不仅会考察其基本实现，还会考察其各种变种，如查找第一个等于目标值的元素、查找最后一个等于目标值的元素、查找第一个大于等于目标值的元素、查找旋转排序数组中的最小值等。这些变种题目需要候选人对二分查找的边界条件和更新逻辑有深刻的理解。例如，一位后端面试者总结中提到了“寻找旋转排序数组中的最小值”，这就是一个经典的二分查找变种问题。字节跳动2024年的面试题目中出现了“搜索二维矩阵”（LeetCode 74）。小红书2024年的面试题目中也有“在排序数组中查找元素的第一个出现位置”（LeetCode 35）以及自定义问题“找到排序好的数组中第一个大于设定目标值的下标位置”。根据知乎专栏文章《刷完这些 Leetcode 题，FAANG TMD BAT 算法必过》的分类，二分查找被列作为一个重要的算法类别。

#### 4.2.7 分治算法

分治算法（Divide and Conquer）是一种重要的算法设计思想，它将一个复杂的问题分解成两个或更多的相同或相似的子问题，直到最后子问题可以简单地直接求解，原问题的解即子问题解的合并。经典的分治算法包括归并排序、快速排序、大整数乘法、Strassen矩阵乘法、最近点对问题等。虽然搜索结果中没有直接突出2023–2024年大厂面试中专门考察分治算法的题目，但分治思想是许多高级算法（如一些树形DP、线段树等）的基础。理解分治思想有助于候选人更好地理解和设计复杂的算法。

#### 4.2.8 并查集

并查集（Union-Find）是一种树型的数据结构，用于处理一些不相交集合（Disjoint Sets）的合并及查询问题。它支持两种操作：查找（Find）：确定某个元素属于哪个子集；合并（Union）：将两个子集合并成一个集合。并查集常用于解决图的连通性问题、网络连接问题、朋友圈划分等。例如，美团2024年春招笔试中就出现了“小美的朋友关系”这道题，其AC代码明确使用了并查集来解决。并查集的实现通常包括路径压缩和按秩合并两种优化策略，以提高效率。掌握并查集的原理和实现对于解决相关问题非常有帮助。

#### 4.2.9 位运算与数学相关

位运算（Bit Manipulation）和数学相关的问题在算法面试中也占有一席之地，虽然出现频率可能不如数据结构和经典算法那么高，但一旦出现，往往需要候选人具备灵活的思维和对计算机底层表示的理解。位运算常用于优化算法性能，或者解决一些特定问题，如“只出现一次的数字”、“比特位计数”等。数学相关的问题则可能涉及到数论、组合数学、概率统计、计算几何等知识，例如判断一个数是否为质数、计算最大公约数和最小公倍数、排列组合的计算、以及一些与几何图形相关的计算问题（如腾讯面试中出现的“一根弦切割圆两部分，给出两部分弓形面积比例，求弦长”）。这类题目考察候选人的数学素养和将数学知识应用于解决问题的能力。CSDN博客中有一篇文章详细解析了斐波那契数列的快速幂运算解法，这既涉及到数学也涉及到算法优化。

## 4.3 难度分析

### 4.3.1 常见难度等级（简单、中等、困难）

国内大厂（BAT、TMD等）在算法面试中，题目的难度通常参照LeetCode等在线判题平台的难度分级，主要分为简单（Easy）、中等（Medium）和困难（Hard）三个等级。从收集到的2023年至2024年的面试真题信息来看，大部分面试题目集中在中等难度。例如，阿里巴巴BU1面试中出现的“mid难度动态规划”题目，以及字节跳动面试中常见的被描述为“基本都是中等难度”的题目，都印证了这一点。美团2024年校招的题目如“不同路径”（LeetCode 63，中等）和“接雨水”（LeetCode 42，中等）也属于这一范畴。简单难度的题目通常用于考察候选人对基础数据结构和算法的掌握情况，例如字节跳动的“判断二叉树的值相等”（LeetCode 100，简单）和美团的“字符串相加”（LeetCode 415，简单）。困难难度的题目在大厂面试中相对较少出现，尤其是在初面或笔试环节，但可能会在后续面试轮次中用于进一步筛选顶尖候选人，或者考察候选人在压力下的解题能力。例如，在一篇LeetCode刷题指南中，作者将某些题目标记为“太难了，先不看”，这暗示了存在一些对普通候选人而言挑战性较大的题目。根据一份对字节跳动面试题的分析，难度题（Hard）只占所有题目的约11.7%，说明大厂算法题对题目难度要求并非一味追求高难度，而是更注重在有限时间内考察候选人的基础和解决问题的能力。

### 4.3.2 LeetCode题目难度与面试题对应关系

国内大厂（BAT、TMD等）的算法面试题与LeetCode等在线判题平台上的题目难度存在高度的对应关系。许多面试题直接来源于LeetCode，或者与LeetCode上的题目高度相似。因此，候选人在准备面试时，可以将LeetCode的题目难度作为重要的参考标准。例如，字节跳动2024年校招面试中的“搜索二维矩阵”（LeetCode 74）、“判断二叉树的值相等”（LeetCode 100）和“求根节点到叶节点数字之和”（LeetCode 129）均为LeetCode上的题目，其难度分别为中等、简单和中等。美团2024年校招面试中的“不同路径”（LeetCode 63），“接雨水”（LeetCode 42）和“字符串相加”（LeetCode 415）也都可以在LeetCode上找到，难度分别为中等、中等和简单。小红书2024年面试中的“会议室问题”（可能指

LeetCode 253. 会议室 II，中等难度）和“在排序数组中查找元素的第一个出现位置”（LeetCode 35，简单难度）同样是LeetCode题目。一篇2020年分析字节跳动算法面试题的文章明确指出，其题目“基本都是力扣的换皮题”。这意味着，LeetCode的难度分级（简单、中等、困难）可以很好地作为评估大厂面试题难度的参考。候选人可以通过大量刷LeetCode题目，特别是中等难度的题目，来熟悉常见的算法题型和解题思路，从而有效提升通过算法面试的概率。同时，也要注意LeetCode上一些会员题或者经典题目的变种，这些也可能成为面试的考察点。

### 4.3.3 不同大厂对算法难度的偏好

虽然国内各大厂（BAT、TMD等）在算法面试中都倾向于考察中等难度的题目，但不同公司在具体的难度偏好和考察侧重点上可能存在细微差异。

**字节跳动** 以其算法面试难度较高而闻名，题目多为LeetCode中等难度及以上，并且喜欢出一些“换皮题”或需要一定思维转换的题目，考察候选人对算法本质的理解和灵活运用能力。其面试不仅关注代码实现，还注重解题思路和优化过程。

**阿里巴巴** 的算法面试题难度也以中等为主，例如其BU1部门明确考察“mid难度动态规划”。阿里面试非常注重项目经验和解决实际问题的能力，算法题往往与项目场景相结合，或者作为考察候选人基本功和逻辑思维的一个环节。

**百度** 的算法面试题据称“最需要好好准备”，可能会涉及一些“没见过”的题目，这表明其题目可能具有一定的创新性或综合性，对候选人的临场应变能力和算法广度有一定要求。

**美团** 的算法面试题难度相对适中，偏向于考察基础算法和数据结构在具体问题中的应用，如动态规划、数组操作和字符串处理，题目多为LeetCode中等及简单难度，但要求候选人能够清晰实现并分析复杂度。

**小红书** 的算法面试题也以中等及简单难度为主，但强调对基础算法的灵活运用和实际问题建模能力，例如其自定义的二分查找变种题。

**腾讯**的面试题中，除了常规的算法题，有时也会出现一些需要较强逻辑思维和问题建模能力的题目，例如涉及游戏规则或实际应用场景的问题。

总体而言，各大厂都将中等难度的算法题作为主流考察点，但字节跳动和百度可能在题目新颖性和思维难度上要求更高一些，而阿里则更侧重算法与项目实践的结合。候选人应根据目标公司的特点进行有针对性的准备。

## 5. 面试准备建议与资源

### 5.1 针对大厂算法面试的刷题策略

成功通过大厂算法面试，离不开系统且高效的刷题策略。首先，**明确目标岗位方向至关重要**，例如是后端开发、算法岗还是前端开发，不同方向对算法和数据结构的侧重点可能有所不同，精准学习比盲刷题更有效。其次，建议从**基础知识入手**，系统学习常见的数据结构（如数组、链表、树、图、哈希表、栈、队列、堆）和算法思想（如排序、搜索、动态规划、贪

心、回溯、双指针、滑动窗口、二分查找等）。LeetCode的LeetBook（原探索栏目）提供了按知识点分类的专项练习，是入门的不二之选，可以帮助理解每个知识点背后的算法和数据结构。在掌握了基础知识后，应按模块精读并结合番茄复盘法进行练习，例如每天集中阅读某个模块的题目和答案，对照总结，记录遗漏的知识点。

刷题顺序上，建议从易到难，从通过率高到通过率低。先刷LeetCode上标记为“Easy”和通过率较高的“Medium”题目，建立信心并巩固基础。对于中等难度题目，要力求理解透彻，做到“多题一解”，即能用同一个算法解决一类共性问题。同时，要重点关注高频题目，因为面试官的题库更新频率可能不高，一些经典题型会被反复考察。可以查阅相关的面经总结或高频题列表，如知乎专栏中整理的针对FAANG、TMD、BAT的LeetCode必刷题。在刷题过程中，不仅要写出正确的代码，还要分析时间复杂度和空间复杂度，并思考是否有更优的解法。此外，模拟面试环境进行练习也非常重要，可以每天假装在面试，快速口答题目，注意语速、表达逻辑和术语解释的完整性。最后，打造自己的知识卡片集，整理题目、考点本质、自己的表达方式以及可能的延伸点，有助于加深理解和复习。

## 5.2 推荐刷题网站与资源（如LeetCode、牛客网）

在准备大厂算法面试的过程中，选择合适的刷题网站和学习资源至关重要。目前，LeetCode是最受国内外面试者推崇的在线判题平台，也是大厂算法面试题的重要来源之一。LeetCode拥有海量的题库，覆盖了各种数据结构和算法类型，并且题目难度分为Easy、Medium、Hard三个等级，方便用户循序渐进地练习。LeetCode的题目大多来源于业界的真实面试题，具有很高的参考价值。平台支持多种主流编程语言，并且提供了Discuss讨论区，用户可以在这里查看他人的解题思路和代码，学习优秀的解法。此外，LeetCode还提供了执行用时分布图表，可以直观地看到自己提交代码的运行效率排名。LeetCode的LeetBook（原探索栏目）提供了按知识点分类的专项练习，非常适合系统学习和入门。

除了LeetCode，牛客网也是国内求职者常用的在线学习和刷题平台。牛客网不仅提供算法题库，还有大量的公司真题、面经分享、在线笔试模拟以及求职交流社区。对于希望了解国内大厂最新面试动态和考察重点的候选人来说，牛客网是一个非常有价值的资源。许多求职者会在牛客网上分享他们的面试经验和遇到的题目，这对于针对性准备非常有帮助。

其他推荐的资源还包括：

- 《剑指Offer》：这是一本经典的算法面试指导书籍，书中总结了大量常见的算法面试题及其解题思路，对于理解算法面试的常见模式和技巧非常有帮助。
- 《算法导论》：这是一本经典的算法教材，内容全面且深入，适合希望系统学习算法理论的候选人。

- **各大公司的官方技术博客或社区**: 例如阿里云开发者社区、腾讯云社区等，有时会分享一些技术文章或面试经验。
- **在线课程和教程**: 如Coursera、edX、B站等平台上有许多优质的算法与数据结构课程。

### 5.3 面试中算法题的解题思路与技巧

在面试中遇到算法题时，清晰的解题思路和有效的沟通至关重要。以下是一些常见的解题思路与技巧：

1. **理解题意，明确输入输出**: 首先，务必确保完全理解题目要求，包括输入数据的格式、范围，以及期望的输出结果。可以主动向面试官确认对题目的理解是否正确，并举一些简单的例子进行说明。
2. **分析问题，提出初步思路**: 在开始编码之前，先花时间分析问题，思考可能的解决方案。可以向面试官阐述你的初步想法，即使不是最优解，也能展现你的思考过程。讨论不同解法的优缺点，例如时间复杂度和空间复杂度。
3. **从暴力解法入手（如果适用）**: 对于一些题目，可以先提出一个暴力解法，然后再思考如何优化。这至少能保证有一个可行的解决方案，并且可以作为后续优化的基础。
4. **考虑常见算法思想**: 根据题目的特点，联想常见的算法思想，如排序、二分查找、双指针、滑动窗口、动态规划、贪心算法、回溯、DFS/BFS等。思考哪种思想可能适用于当前问题。
5. **画图或举例辅助思考**: 对于一些复杂的问题，可以通过画图、举例或者列出关键步骤来帮助自己理清思路。这也有助于向面试官清晰地展示你的思考过程。
6. **分步实现，模块化代码**: 在编码时，尽量将代码模块化，将复杂的功能分解成若干个小函数或步骤。这样不仅代码结构更清晰，也便于调试和修改。
7. **注意边界条件和异常处理**: 在编写代码时，要特别注意边界条件的处理，例如空输入、单个元素、最大值、最小值等情况。考虑代码的鲁棒性，处理可能的异常情况。
8. **测试你的代码**: 完成编码后，不要急于提交，先用一些测试用例来验证代码的正确性，包括常规用例、边界用例和特殊用例。向面试官解释你的测试思路。
9. **分析时间复杂度和空间复杂度**: 在代码通过测试后，主动分析算法的时间复杂度和空间复杂度，并讨论是否有进一步优化的空间。
10. **保持沟通，积极互动**: 在整个解题过程中，与面试官保持良好的沟通。如果遇到困难，可以请求提示；如果有新的想法，可以及时提出讨论。展现出积极思考和合作的态度。