



大学生 如何攻克算法面试

主讲人 苏铭彻
BitTiger 联合创始人



主讲人介绍

苏铭彻

BitTiger联合创始人，旅美科协硅谷分会副会长

毕业于卡内基梅隆大学机器人研究所，研究方向：计算机视觉

曾就职于WalmartLabs和Groupon从事大数据系统架构的研发工作。

1

CHAPTER1

算法和数据结构在软件工程师求职的重要性

2

CHAPTER2

如何准备算法和数据结构面试?

3

CHAPTER3

面试算法题目的公式以及考查形式

4

CHAPTER4

数据结构和算法知识点梳理

5

CHAPTER5

典型面试题讲解

CHAPTER1

数据结构和算法的重要性

- 程序 = 数据结构 + 算法
- 实际的面试考量

程序 = 数据结构 + 算法

- 数据结构是对业务数据抽象出来的一种模型，而算法则是解决业务问题的思维方法论
- 是否有高度的抽象能力和是否具有计算机思维是高水平程序员和一般码农的分水岭
- 程序 = 算法 + 数据结构
软件 = 程序 + 软件工程
软件公司 = 软件 + 商业模式
- 往往软件工程的质量直接决定软件的质量，商业模式的优劣决定了一个公司的成败



1. 数据结构和算法是大学生基础课：基础不牢，地动山摇
2. 应届生自身经验不足且参差不齐，以项目经验为标杆并不好考察一个面试者
3. 大公司：对于应届生更加注重的应该是个人能力和发展潜力如何，能够容忍你一个学习期，一定程度上弥补经验的不足；
4. 数据结构和算法比较通用，而且面试官心里已经有预期，答案的正确性比较好考察。

CHAPTER2

如何准备数据结构和算法面试

- 刷题方法的误区
- 科学的刷题阶段
- 刷题语言的选择
- 刷题的自我联系

上来就刷，刷一道是一道
题海战术，让我再刷100道
死记硬背，不求甚解
缺乏总结，没有变通

题海无涯、回头是岸

初期：熟练掌握各种基本工具

中期：熟练各种类别题目解题思路

后期：做新题，总结老题，一题多解，多题同解

面试前：做面经，别纠结

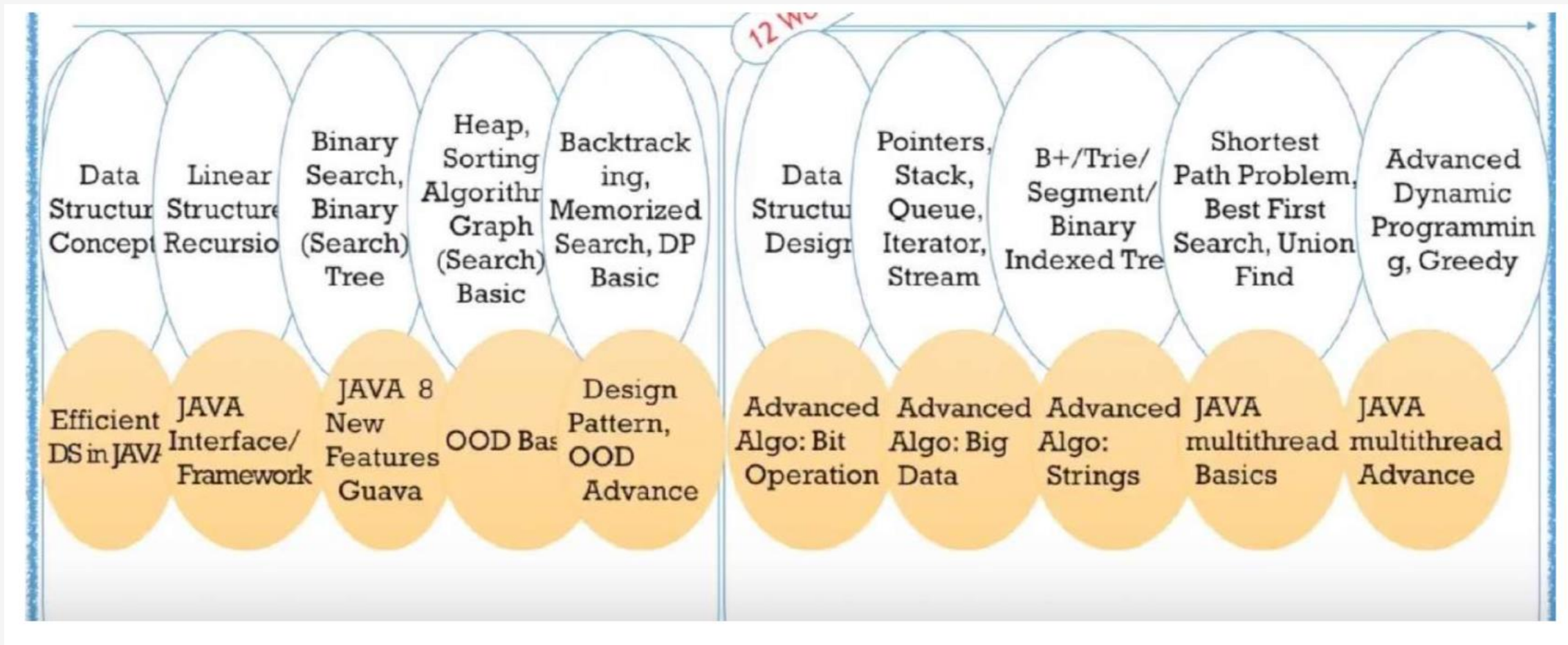
按难度分类，先做简单题，广度优先

按类别分类，广度优先

按“类别”分类，深度优先

按公司分类，因公司而异





- 在线评测系统：
- <https://leetcode.com/> <https://www.nowcoder.com>
- 白板写题：外企比较常见
- 白纸写题：国内比较常见
- 电脑写题：现场coding
- 把题给别人讲清楚才是检验刷题成功与否的唯一标准



CHAPTER3

面试算法题的公司以及考查形式

- 互联网公司判断
- 真实面试考查形式
- 算法在面试中的比例

互联网公司盘点

- 外资公司: **Google/Microsoft/Ebay/Hulu/Amazon/Airbnb/Snap/Vmware/Linkedin**
- 中国互联网大厂: **百度、阿里巴巴、腾讯、头条、美团、滴滴、京东、网易、搜狐...**
- 知名互联网公司: 知乎、饿了么、KEEP、拼多多、小红书、摩拜、ofo...
- 新兴行业: 无人车、Fintech (互联网金融)、计算机视觉等

面试考查形式

在线评测： 数据和算法基本概念

电话面试： 算法题思路/在线写题

现场面试： 白纸写题/白板写题

算法和数据结构所占比重

外企考查比重大： Google等几乎全是算法

国内大公司比例比较高： 35-60%

应届生比重大： 有工作经验的面试者项目比例比较大

取决于面试官，名校出身的面试官喜欢考算法

CHAPTER4

数据结构和算法考点梳理

- 基本数据结构
- 算法：时间和空间
- 实战练习大纲

基本数据结构

- 哈希表 Hashtable, 及其背后所代表的哈希的思想是如何应用的不同问题的, 如果只能用数组来实现, 要如何实现
- 树 Tree, 树的构建(trie), 遍历和修改。遍历包括 BFS 和 DFS (前序中序后序), 及其对应的思想。如果了解至少一种平衡树 (及实现方式) 就更好了。
- 图 Graph, 图的三种表达方式及优缺点, 基本的遍历算法 (还是 BFS 和 DFS), 知道这些算法的复杂度以及权衡利弊。如果了解一些如 Dijkstra 和 A* 算法就更好了。
- 其他的数据结构: 堆栈队列链表

算法：时间和空间

- **分而治之**：有点类似“大事化小、小事化了”的思想，经典的归并排序和快速排序都用到这种思想，可以看看 [Search a 2D Matrix II](#) 来理解这种思想。
- **动态规划**：有点类似数学中的归纳总结法，找出状态转移方程，然后逐步求解。 [309. Best Time to Buy and Sell Stock with Cooldown](#) 是理解动态规划的一个不错的例子。
- **贪心算法**：有时候只顾局部利益，最终也会有最好的全局收益。 [122. Best Time to Buy and Sell Stock II](#) 看看该如何“贪心”。
- **搜索算法**（[深度优先](#)，[广度优先](#)，[二分搜索](#)）：在有限的解空间中找出满足条件的解，深度和广度通常比较费时间，二分搜索每次可以将问题规模缩小一半，所以比较高效。
- **回溯**：不断地去试错，同时要注意回头是岸，走不通就换条路，最终也能找到解决问题方法或者知道问题无解，可以看看 [131. Palindrome Partitioning](#)。

参考实战练习大纲（选自网络）：链表

1. 链表的必备知识要点(包括基础知识、刷题中使用的STL等知识)
2. 链表逆序(LeetCode 92,206. Reverse Linked List 1,2)
3. 求两个链表的交点(LeetCode 160. Intersection of Two Linked Lists)
4. 链表的节点交换(LeetCode 24. Swap Nodes in Pairs)
5. 链表求环(LeetCode 141,142. Linked List Cycle 1,2)
6. 链表重新构造(LeetCode 86. Partition List)
7. 复杂的链表复制(LeetCode 138. Copy List with Random Pointer)
8. 排序链表合并(2个与多个) (LeetCode 21,23 Merge Two(k) Sorted ListsLeetCode)

参考实战练习大纲（选自网络）：栈、队列、堆

1. 栈、队列知识要点与实现(数组、链表)
2. 使用队列实现栈(LeetCode 232. Implement Queue using Stacks)
3. 使用栈实现队列(LeetCode 225. Implement Stack using Queues)
4. 包含min函数的栈(LeetCode 155. Min Stack)
5. 简单的计算器(栈的应用)(LeetCode 224. Basic Calculator)
6. 堆(优先级队列)知识要点与实现
7. 数组中第K大的数(堆的应用) (LeetCode 215. Kth Largest Element in an Array)
8. 寻找中位数(堆的应用)(LeetCode 295 Find Median from Data Stream)

参考实战练习大纲（选自网络）：贪心

1. 贪心算法知识要点，刷题必备的STL知识
2. 贪心题目1(LeetCode 455. Assign Cookies)
3. 贪心题目2(LeetCode 402. Remove K Digits)
4. 贪心题目3(LeetCode 134. Gas Station)
5. 贪心题目4(LeetCode 135. Candy)
6. 贪心题目5(LeetCode 502. IPO)
7. 贪心题目6(LeetCode 321. Create Maximum Number)
8. 贪心题目7(codeforces 582A GCD Table)

参考实战练习大纲（选自网络）：树与图

1. 树与图的数据结构与基本算法
2. 树遍历的回调函数实现，并使用自动机概念实现非递归树前、中、后遍历
3. 树与链表的转换(LeetCode 114. Flatten Binary Tree to Linked List)
4. 最近的公共祖先(LeetCode 236. Lowest Common Ancestor of a Binary Tree)
5. 树的层次遍历应用(LeetCode 199. Binary Tree Right Side View)
6. 树的改造(LeetCode 117. Populating Next Right Pointers in Each Node 1,2)
7. 图的复制(LeetCode 133. Clone Graph)
8. 图的搜索与应用(LeetCode 207.Course Schedule)

参考实战练习大纲（选自网络）：二分查找、二叉排序树、位运算

1. 二分查找、二叉排序树的知识要点
2. 数组的二分查找(LeetCode 33,81 Search in Rotated Sorted Array 1,2)
3. 区间二分查找(LeetCode 34. Search for a Range)
4. 排序链表转换为二叉排序树(LeetCode 109. Convert Sorted List to B- Search Tree)
5. 二叉排序树的遍历与改造(LeetCode 538 Convert BST to Greater Tree)
6. 二叉排序树中的第K大的数(LeetCode 230. Kth Smallest Element in a BST)
7. 位运算的知识要点
8. 使用位运算表示集合(LeetCode 78. Subsets)
9. 位运算应用题目(LeetCode 136,137,260. Single Number1,2,3)

参考实战练习大纲（选自网络）：搜索

1. 深度优先搜索与广度优先搜索算法
2. 深搜题目 (LeetCode 200. Number of Islands)
3. 深搜题目 (LeetCode 473. Matchsticks to Square)
4. 深搜题目 (LeetCode 491. Increasing Subsequences)
5. 广搜题目 (LeetCode 126,127 Word Ladder 1,2)
6. 广搜题目 (LeetCode 417. Pacific Atlantic Water Flow)
7. 广搜题目 (LeetCode 407. Trapping Rain Water II)

参考实战练习大纲（选自网络）：动态规划

1. 动态规划知识要点
2. 动态规划题目1(LeetCode 120. Triangle)
3. 动态规划题目2(LeetCode 53. Maximum Subarray)
4. 动态规划题目3(LeetCode 198,213. House Robber 1,2)
5. 动态规划题目4(LeetCode 322. Coin Change)
6. 动态规划题目5(LeetCode 72. Edit Distance)
7. 动态规划题目6(LeetCode 174. Dungeon Game)
8. 动态规划题目7(codeforces 711C Coloring Trees)

参考实战练习大纲（选自网络）：复杂数据结构

1. Trie树的构造与基本算法
2. Trie树的构造 (LeetCode 208. Implement Trie (Prefix Tree))
3. Trie树的应用 (LeetCode 212. Word Search II)
4. 并查集的基本算法
5. 并查集的应用 (LeetCode 547. Friend Circles)
6. 线段树与树状数组
7. 线段树与树状数组的应用(LeetCode 307. Range Sum Query – Mutable)

CHAPTER5

典型题目讲解

两数之和
快乐数
海量数据的Top N问题

两数之和

题目介绍: Input: numbers={2, 7, 11, 15}, target=9, 找到 $a + b = t$ 的序列对 (a,b)

- 1) 两层循环 $O(n * n) \Rightarrow$ 排序 $O(n * \log n)$
- 2) 一层循环 $O(n)$, 内存占用 $O(n)$
- 3) 结果如何去重
- 4) 扩展到Three SUM问题

Happy Number: 快乐数

题目介绍：判断一个数是否快乐数：定义一种快乐数，就是说对于某一个正整数，如果对其各个位上的数字分别平方，然后再加起来得到一个新的数字，再进行同样的操作，如果最终结果变成了1，则说明是快乐数，如果一直循环但不是1的话，就不是快乐数。数字19就是一个快乐数。

$$1^2 + 9^2 = 82$$

$$8^2 + 2^2 = 68$$

$$6^2 + 8^2 = 100$$

$$1^2 + 0^2 + 0^2 = 1$$

步骤1：明确题目含义和函数输入和输出

1. 写一个boolean型的函数：`bool isHappy(int n)`
2. 输入可否为负数 或者为小数，明确类型应该为integer

步骤2: 讲解算法思路 (非实现) : 依据需求选择数据结构

Algorithm

```
• public boolean isHappy(int n) {  
    while (n != 1) {  
        • 计算sum  
        • if (sum exists) { // 查找sum  
        •     return false;  
        •     }  
        • 保存sum  
        • n = sum;  
        • }  
    }  
    • }  
    • return true;  
    • }
```

- Store sum + Check if sum already exists
- -> Hash
- -> HashSet

步骤3：代码实现

Code

```
• public boolean isHappy(int n) {  
    Set<Integer> s = new HashSet<Integer>();  
    • while (n != 1) {  
        • int sum = 0;  
        • while (n != 0) {  
            • int mod = n % 10;  
            • sum += mod * mod;  
            • n /= 10;  
        • }  
        • if (s.contains(sum))  
            • return false;  
        • n = sum;  
        • s.add(sum);  
    • }  
    • return true;  
    • }
```

步骤4：找测试数据，验证代码正确性

一般来说，测试用例要涵盖以下情况：

- 小规模、可以手算的数据。用于快速检验程序的正确性。要注意规模不能太小以至于过分简化问题，比如对于图论题，只有2、3个点的图就太小，一般要取5、6个点。
- 大规模，但仍有规律、可以手算的数据。也是用于检验程序的正确性。如果程序算出来的结果跟手算的不同，不要急着debug，还需要考虑一下是不是手算错了哦~
- 边界和极端情况。一方面检验程序的正确性，另一方面保证复杂度不超过限制。比如二叉树退化成一个链的情况。
- 大规模随机数据。用于保证程序不会崩溃，以及复杂度不超过限制。

输入格式不正确的情况，在OJ上一般可以忽略，虽然在实际应用中必须考虑。

步骤5：分析算法时间复杂度和空间复杂度

Complexity

- Time $O(1)$
- Input positive integer
- $1 \sim 2^{31} - 1$
- $2^{10} = 10^3$
- $2^{20} = 10^6$
- $2^{30} = 10^9$
- Max : $2 * 10^9 \rightarrow 10$ digits
- (sum)max = $9^2 * 10 = 810$
- Space $O(1)$

算法题中穿插的面试问题

- hashset /treeset的区别;
- 为什么选择hashset?
- java最大的integer是多大?
- 如何估计 $2^{31}-1$ 的大小?
- 补码和反码

海量数据求topN问题

题目介绍:

一个文件10行url, 单机4G内存, 求出现次数最多的10个URL

- 1) 考虑如何存储? 占用内存空间
- 2) 查看最小堆
- 3) 考察归并排序, mergeSort
- 4) 考察大数据思维

拉勾校园算法竞赛，给简历加点料

【活动受众】

- 19/20届计算机相关专业在校生（或者你自学了编程，要来小试牛刀~）
- 个人或组团报名均可
- 每组不超过5人

【奖项设置】

- 第一名小组：1500元奖金及证书
- 前三名小组：拉勾独家名企直推通道
- 前五名小组：名企Openday活动1次

【活动时间】

6月9日 13:00—17:00

【活动地点】

北京大学
创新创业中心地下一层 交流区



扫码查看活动&报名



Thank You

