

# 高效准备算法面试

无隅

# 自我介绍

无隅

- 毕业于北京邮电大学
- 现就职于亚马逊中国研发中心 SDE
- 擅长于Java/Python/数据结构与基础算法
- Interest: 分布式系统, 机器学习

# 提纲

- 面试流程解读
- 算法面试考察点解读
- 如何解答算法面试问题
- 如何准备算法面试

# 面试流程

- 筛选出你感兴趣的公司，最好能找到认识人内推
- 准备简历 && 提交简历
- HR联系
- 电话面试 / 在线面试
- 现场面试 (onsite interview)
- offer谈判

# 找到好工作的关键

- 时间 – 规划、安排与提早准备

- 实力

硬实力

软实力

- 时间甚至比实力更重要

# 校招找工作时间安排

假设2018年为研究生二年级或者大三，2019年毕业

2017年7月到3月准备实习面试

2017年10月到2018年3月 实习面试

2018年5月 - 8月实习

2018年7月 - 10月准备全职面试

2018年8月到11月 面试

2019年工作

# 关于简历准备

- 合适的简历长度

建议校招的同学把简历长度保持在一页以内。

对于有比较久工作经验的面试者可以将简历长度调整到1.5 - 2页

- 内容写什么：个人基本信息，教育背景，专业技能，项目经历与工作经历，获得的奖项与奖励
- 建议不要写什么：兴趣爱好，个人评价，与技术背景无关的经历
- 注意不要有错别字，尤其是个人联系方式
- 统一字体格式，建议最好用pdf格式提交

# 简历准备 - 项目经历与专业技能的小建议

- 项目经历

1. 拿出2 - 4 个最具代表性的项目，不要添加太多项目
2. 最好能用数据说话，比如性能提高了多少，增长了多少新用户，部署流程简化了多少
3. 点出项目的技术亮点与实现方案

- 专业技能

1. 关于软件与框架

请保守一些（了解过 / 使用过 / 研究过），以免给自己挖坑

请不要写熟练使用word / excel / ppt

2. 关于语言：还是请保守些，可以列出所有你常用的，  
但请标注程度或者使用频率百分比



# 面试：清楚面试过程，才能对症下药

- 行为面试 （自我介绍与项目介绍） 5min
- 解决问题 （算法题） 25 - 30 min
- 对面试官的提问 5min

## 面试的意义

考察一个面试者是否有胜任该职位的能力

# 面试官视角下的求职者

- 求职者是未来的同事，一起工作的人
  - 求职者的思维方式与工作模式是否符合公司价值观与工作流程
- 求职者的代码可读性与专业能力
  - 是否在code review中需要浪费大量的时间
  - 是否可以胜任日常工作
- 求职者的编程习惯
  - 是否遵循基本的编程原则
  - 是否有良好QA的意识

# 面试考察点

- coding能力与语言熟练程度
- 良好的代码风格coding style
- 算法分析与实现能力
- 良好的测试覆盖
- 理解与沟通能力

# 同学们常遇到的问题

- 熟练程度不够，没有bug free的能力
- 碰到原题欣喜若狂，碰到新题垂头丧气
- 思路不够清晰，基础不够扎实
- 埋头写题，全程没有交流
- coding style很糟糕，代码可读性差

# 饥人谷算法班，给你的远远不止这些

- 面试涉及的算法基础知识
- 带你刷题
- 教你写简历
- 给你模拟面试，然后给你改进的建议
- 任何与找工作有关的事情

# 算法面试问题解答流程

- 明确问题：与面试官交流明确要解决的问题
- 举例分析：用一些例子来尝试解决问题
- 描述大体思路：让面试官了解你的思维过程
  - 先给出暴力解法：不要担心效率，Done is better than perfect!
  - 尝试优化：BUD原则，头脑风暴
  - Walk through
- 代码实现：写干净整齐的代码，注意抽象模块化
- 给出测试用例
- 分析算法复杂度，回答问题

# 怎么优化和解决问题

- BUD原则
  - Bottlenecks
  - Unnecessary Work
  - Duplicated Work
- 模拟
  - 尝试从基本例子入手
  - 简化和抽象问题，从特殊推广到一般
- 头脑风暴：利用数据结构特性与边界条件



# 如何准备算法面试

- 明确考察范围
- 分类、总结、分清主次
- 提高代码能力
- 提高coding style
- 一类题型有多少种变种
- 一个算法有哪几种题型
- 不需要做完所有题目

# 核心考察知识范围

- 知识点范围
- 题目资料来源
  1. leetcode
  2. 面经

# 高效刷题之能力提高

- 如何提高代码能力

1. 做题的时候，把题目记下来，然后开始做题，做完题提交如果有错，记录下你的错误
2. 每一次做这道题出现的错误都请记录下来
3. ac通过后请记录思路

- 如何提高代码风格coding style

1. 项目代码风格参考手册

java: <https://google.github.io/styleguide/javaguide.html>

阿里巴巴Java开发手册 - 编程规约

JavaScript: <https://google.github.io/styleguide/jsguide.html>

python: <https://google.github.io/styleguide/pyguide.html>

c++: <https://google.github.io/styleguide/cppguide.html>

2. 面试代码风格：命名、空格、缩进、括号等

# 如何检验自己已经准备好了

- 除了能ac外还会考虑性能问题
- 良好的边界条件处理 corner case
- 在白纸白板上可以流畅地完成题目
- 量化指标：
  1. 通过一个题目的平均提交次数在2次左右
  2. 提交的代码能够击败90%以上

# 遇到难题或者没见过的题怎么办

- 要跟面试官交流思路，即便不会也要把你的思考过程展现出来
- 不可能所有题目都做过所以不会很正常，要保持良好的心态
- 交流题目具体内容
- 了解题目要点与限制条件，说说自己的大致思路
- 要与面试官多交流，能写多少写多少，千万不要不写

# 关于面试官你不得不知道的事情

- 很多时候面试官主观感受是大于客观评价的
- 交流第一，做题第二
- 思路第一，写题第二
- 与面试官争论一个点是否正确意义不大
- 情绪与状态请不要受到上一轮面试影响
- 面试官给的反馈主观因素占比很大

# 面试准备建议

- 夯实基础，别做难题
- 是面试而不是考试
- 着重总结，理解而不是背诵
- 时间要用在重要的地方

# 夯实基础，别做难题

- 不要花时间攻关难题
- 把时间花在如何做到bug free和如何提高编程速度上
- 集中突破简单和中等难度的题目上
- 理解数据结构与算法背后的故事



# 是面试而不是考试

- 面试是为了招聘合适的人一起工作
- 面试是双方选择的过程
- 和面试官愉快交流，一块合作解决面试问题
- 证明自己的实力，但别去证明面试官的实力

# 着重总结，理解而不是单纯的背诵

- 主要学习的是思维方式和分析技巧
- 要理解数据结构与算法的内在原理，熟练地给出模板实现
- 善于总结一类题目的模版与内在原理  
而不是某个题的解法

# 时间要用在重要的地方

- 不要把时间浪费在那些基本不会考你又很心虚的内容
- 比如KMP，红黑树，线段树，AVL以及很复杂的题目上

# 饥人谷算法班，给你的远远不止这些

- 面试涉及的算法基础知识
- 带你刷题
- 教你写简历
- 给你模拟面试，然后给你改进的建议
- 任何与找工作有关的事情

Q&A