

别问LZ为什么在微型startup干,问就是失业天快用完了,没办法挑了一个startup开始OPT. (世界杯看球给耽搁了18年秋招)但是既然开始干了,就抱着为公司做贡献,自己学东西的心态干吧,工资刚刚够吃.

但是**就是这一个工作经验,还是弥补了我简历上没有美国本土公司工作经验的缺憾.人生那么长,一时得失不必在意,明白自己应该从工作中获得什么就够了.这也是我最想说的一点,从我身边亚麻社招的惨况来看,**又OA1/2做完挂了的,又Video面的偏难的,所以社招有时候不一定比校招面的难.如果在校还有时间的同学,拿不到面试(我也是这么过来的),可以考虑去当地的start up打打工,争当

相信我,又美国本土的工作经验,简历不容易被据,当然你得作出贡献,平时多记录怎么调出bug,怎么处理deadline和time management,这点还挺重要的,bq很多会问到.

导读 | 版块 | VIP瞬间解锁 | 应用中心 | 功能中心 | 网课 | 攻略 | 免米搜索 | 关于 | 常见问题FAQ |

LZ刷题一直是200 到300, 但是很多题能很熟练的多解法写出来。300题之前以及比较经典的过了可能得有6遍. 刷不动题的就抄答案, 然后理解, 按照遗忘曲线复习, 自己写出来, 再想.

另外很多题其实就是一些简单题的改写, 比如LZ BB面挂的flatten BST, 就是dfs框架上保存postorder exit信息, 在改写的时候不要慌, 好好想透了再下笔, 达到你保证下次碰到就能拉出框架的地步.

再来就是熟练度和该不该记答案, 说到底, 面试当场smart的人能够凭借一点思路然后套面试官的话一点点把最后答案构建出来. 但是LZ自知不属于, 所以老老实实的一遍一遍刷题, 每刷一遍熟悉以下这题分几个main step, 这也是你可以在拿到题之后开始解题之前说思路的参考. 然后对于2遍不能AC的题, 一定要用英文写下bug在哪, 可以当作写完code之后拿出来和面试官讨论bug在哪, 也提醒自己下次不犯.

3. 答题节奏

提供一些模版句, 我觉得和考托福口语一样, 安静确实有点尴尬. 想思路的时候 可以说let 's mind storming, find what data structure and algorithm we need. 如果5min内不能拉出大框架, 一定记住问hint, 否则在错误的路上越走越远, 面试官也帮不了你. 不要觉得问hint很丢面子, 其实如果面试官一开始出的题就很非常规, 非套路(当年面gg跪就是这么跪的), 不要hint除非大神一般硬想想不出来. 因为作为面试官, 要么你写code说思路他来挑错找corner case. 要么就是你问hint, 然后和面试官一起讨论得出解法. 所以要么你很牛b. 一眼看穿解法, 要么你就只有半截思路, 尝试和面试官找到所有思路.

不要长时间卡在不通的思路上,多问hint,做出来肯定是比没做出来要强;

不要长时间自言自语, 没人喜欢和一个自说自话的合作;

不要没有思路的时候, 急于写代码, 一定是先确认思路再问面试官可不可以implement;

如果卡在中间的code, 要大改动就是最忌讳的了, 如果能不大改写下去, 那么就继续, 过后再说优化. 否则改来改去面试官可能觉得你考虑问题不全面.

-follow up

我面试也是在提醒下才能回答上follow-up, 感觉follow-up就更加注重和面试官的交流, 解释代码的同时告诉他你的思路.

面试说到底还是同类型的题能做到第一遍写个大差不差(45度炸.董哥粉丝走一波,陪我走过无数个失眠焦虑的夜晚).

至于系统设计,我个人感觉就是先从简单的case拓展到复杂,不要上来就说应该怎么设计,而应该先说为什么,瓶颈在哪里.一般先说naive 的方法瓶颈,再说你的solution. 然后面试官的问题一定要确认清楚再回答.

4. BQ

不确定BQ对面试的影响,估计是有一定作用的,多记录工作的debug经验,多记录怎么learn & teach,多记录project里面学到的东西概念,注重描述的时候总分总,不啰嗦,简明扼要又要又一定的细节.

本主题由 清道神君 于 2019-3-19 15:36 审核通过

快速回复

〇 评分

参与人数 22	大米 +31	理由	3 4
joker468	+ 1	赞	
BZydNpRW	+ 2	很有用的信息!	
hanchancs	+ 2	给你点个赞!	
andyz candyz	+ 1	给你点个赞!	
• o isabellalalalal	+ 2	给你点个赞!	









