



我的人缘 0

2 主题 | 46 帖子 | 523 积分

发消



分享帖子到朋友圈

wkroach 4 天前 | 只看该作者 ▾

1 楼 电梯直达 □ ↗

 本楼: 100% (15) 0% (0)
 全局: 98% (95) 1% (1)

2019(10-12月)-CS硕士+短暂实习或全职不超过3个月 | 网上海投|Texas 码农类General全职@

大家好，楼主刚刚结束了2019 new grad SWE的找工，过程中领悟了许多关于找工、面试、心态调节等等的一些心得体会和一些实用的技巧，于是打算把它们都总结一下，既是为了保留住这些宝贵的经验教训，也是为了之后的找工者能更加从容，顺利。地里大家无私分享的面经和找工经验帮了楼主很多很多忙，希望这篇心得总结也能够帮助到那些仍在找工，以及未来需要找全职或实习的同学们，能给你们一些灵感。也借此感谢找工岁月中所有帮助过我的人，感谢同我并肩作战，给我鼓劲加油的所有战友！

万字长文，所以做了一份目录，方便大家阅读

目 录

1. 简历
 - a. 简历修改
 - b. elevator pitch 模版
 - c. 经历介绍模版
2. 职位投递
 - a. 信息来源
 - b. 投递方式
 - c. 简历投递策略
3. 面试
 - a. 面试tips
 - b. 面试沟通模版
 - c. [刷题](#)
 - d. 面经
 - e. Behavior question
 - f. [system design](#)
 - g. 面试规划

简历

简历，elevatorpitch(自我介绍)，和项目介绍，都是在投递职位，或者跟HR在CF/GHC中聊天会用到的，决定了我们能够拿到多少面试。初期拿到的面试越多，对自己的自信 and 良好心态的促进越大，更多的面试也会大大影响最后我们上岸的时间和可能性，而不是白白浪费很多时间和机会。所以，在正式找工作投简历参加CF/GHC前，强烈建议把这些都准备充分，一劳永逸。

简历到底应该怎么写？

简历很重要，虽然当我们开始找工作的时候，个人的背景，项目经历已经差不多定型了，但同样的经历不一样的写法结果差异也会很大很大，哪怕是平乏无味的内容，好好地把格式和内容润色一把也可以效果很好，一把抓住HR眼球，拿到面试。

大方向：

1. 在不骗人、不无中生有的条件下，尽最大可能让经历看起来很厉害
2. 结构很重要，最厉害的项目/经历放在简历最前面，经历里最厉害的impact和总结性质的project描述放在最前面
3. 加粗很重要，把前沿的、工业界常用的skills在经历里加粗，对筛简历很有用
4. 面试官更倾向于看到跟招聘职位/自己team有关联的tech和field，比如你找无人驾驶就最好有C++和ml相关的project经验，找hedge fund就最好有C++和system相关的经验，所以有较强关联的经验，针对特定公司，要放在最前面，把最吸引他们的点加粗。
5. 内容建议顺序: Education+experience/projects+skills+others
6. 排版+字体+格式都很重要，选模版记得不要让最重要的项目经历都堆在简历的左边或右边，这不太符合人的阅读习惯，哪怕这样看起来很好看

细节：

1. 总结自己目前的项目经历，尽量详细的写下都做过什么，以及达到了什么样的impact。
2. 尝试放大impact，把impact放在更大scope的角度来分析对team/公司/用户产生了什么样的benefit。
3. 量化impact，提升了多少收益/performance/throughput，降低了多少成本/memory/time/人力/latency，最好用数字和百分比表示出来，一般用了数字的话可以不用加粗，因为已经比较明显了。
4. 梳理每一条bulletpoint，如果一段经历里有几个project，把主要的impact最大的project放在前几条来说，最好在第一时间把project的名字加粗，并把最大的impact写出来。
5. 每一条bulletpoint，这样写：
 - a. 做了什么东西，用了什么样的前沿的tech，这个东西有什么结果，达到了怎样的impact。
 - b. 顺序未必要严格按照这个来，但是东西都要有，强调impact。
6. 如果一段经历，比如说实习里做的project都很水怎么办？写不了几个bp怎么办？可以这样写：
 - a. 列出这段经历里做的几个project A, B, C，它们最好有一定关联

- b. 选出impact最大的A, 然后把B, C的内容和impact也算作是A的
- c. HR/面试官问简历都不会很细地去核实每一条, 所以只要都是自己做过的东西, 适当改编一下言之有理即可, 不用太死板

Elevator pitch 模版

1. 介绍自己的姓名, 目前教育/工作情况 (一句话就够)
2. 介绍自己有几段实习/工作经历, 对于每段工作经历, 按照如下模版准备:
 - a. 在某年某每月, 在**公司实习/工作了多久
 - b. 职位是什么, 所在的team是干嘛的, 主要负责哪个业务 (infra/machinelearning/product), team对公司的impact是什么
 - c. 我在这段时间里的project是什么, 用了什么tech, 有什么样的impact
3. 很短的介绍其他的经历, 比如比赛, 或者其他project
4. 最后, 说I am interested in the ** job in ** company
5. Optional: Do you want to know more about any experience I have, I can explain it more detailly.

经历介绍模版

经历介绍本质就是一个比较短的presentation

1. 介绍project的背景, 要解决怎样的一个问题, 这个问题的原因是什么, 产生了什么不好的结果
2. 已有的方案是什么样的, 为什么不能够解决这个问题
3. 我的project能够干嘛 (impact), 为什么, 以及如何解决这个问题
4. 这个project在high level上是如何工作的, 用了什么tech, tool, 技术
 - a. 在细节上, 我做了什么A (遇到的问题A, 给出了怎样的solution A)
 - b. 在细节上, 我做了什么B (遇到了问题B, 给出了怎样的solution B)
 - c.
5. 最后这个project的结果如何 (上experiment/production), 上production的话效果如何
6. 如果没有上production, 做了怎样的experiment证明了这个project的impact

职位投递

简历准备好了, 就该投出去了。投简历还是越早投递越好。

职位信息来源

[linkedin](#), 一亩三分地, 公司官网, 猎头

公司官网的信息是最早的, 一般比LinkedIn要早, 所以对于针对的特别想去的公司, 每周隔三差五地刷官网很重要。LinkedIn也是, 对于要投简历的公司, 提前开启职位提醒, 然后每天看一下有没有update。

信息搜集一定要早, 一旦进入找工作的高峰期 (9月10月), 拿面试几率大大降低, 越早投递内推, 拿到面试几率越大

投递方式

熟人内推=猎头 > LinkedIn 学长学姐内推 >= 网投 > 一亩三分地内推

熟人的有些内部资源是其他渠道难以替代的。LinkedIn找学长学姐推的几率比其他人的几率大很多。部分公司, 尤其是金融类的公司, 网投是完全可以的, 不过认识靠谱的猎头的活也是很方便的, 信息的反馈会很及时。一亩三分地内推同样, 越早发邮件联系, 越早有机会拿到内推, 晚一点点都会爆满。

关于Career fair (CF), 这里说一下Rice的情况(仅针对2019 fall, 2020+的情况概不负责, 学弟学妹不要因为楼主一家之言就改变自己的判断-。-), Rice 9月会有Careerfair, 10月初会有针对CS系的careerfair, 叫CS mixer。其实Rice的大CF的科技公司不是很多, 但是, 但是, 但是, Rice会有些小公司或者非tech公司来招人, 比如斯伦贝谢, 今年还有个小start up叫extro ho, 斯伦贝谢是Rice这边比较稳的上岸公司了。CS mixer的科技公司很多, 但很少会让你现场投简历, 只让我们提前在handshake上传简历, 所以有多少人通过这个拿到面试就不得而知了, 楼主拿到了1个nuro的面试。大公司在CS mixer发面试的据我所知还有apple

简历投递顺序和周期

1. 列出所有想去的公司名字
2. **按照公司拿offer难度和面试难度进行梯度分级 (先上岸把心态调整好相当重要)**
3. 每个级别的公司, 取最想去为基准进行排序, 开始搜集投递方式
4. 有内推, 就立马找内推, 如果没有熟人内推, 找LinkedIn
5. 由于LinkedIn和一亩三分地回复一般会很慢很慢, 要看这个公司的内推和网投是否冲突, 不冲突可以先投, 否则先等3天
6. 如果实在找不到内推, 就网投
7. 每周花一个特定时间, 重复1~6的工作, **不要因为拿的面试多就放松了投小公司, 上岸是找工初期首要目标!**
8. 对尚未放出职位的公司标记LinkedIn提醒, 每天刷一刷, 如果有新职位, 立马找内推 (速度! !)

面试

面试的本质, 既是面试官在选自己未来一起共事的, 一同解决问题的同事, 也是面试者在考察自己将要参与工作的公司和同事, 而不是一个单方面对面试者的考试

面试是一个动态, 交互的过程, 就和dating一样, 是双方通过共同解决一个算法问题/系统问题/设计问题, 以及互相聊天, 了解双方的背景, 来回答这么几个问题的答案:

1. 公司/组需要解决什么样的问题, 期待有什么能力, 技能, 经验的员工, 这个面试者是否满足这个要求, 以及双方一起沟通合作解决问题是否会顺畅愉快----->technical面试(算法题/system design等等)
2. 这个面试者是否能够融入公司的氛围, 文化, 他是怎样的一个人, working style如何, 是不是很professional----->BQ
3. 这个面试者想要做什么样的工作/项目, 对什么样的project有兴趣, 喜欢什么样的工作氛围, 想从这份工作中得到什

么，对自己的未来职业发展有怎样的规划和期待-----> BQ

4. 公司的资源、项目、工作环境等等能否满足这个面试者对这份工作的期待，他是不是真的喜欢这个公司并愿意积极参与到工作之中-----> BQ

而这些问题答案，决定了一个最终问题的答案：双方是否match，如果面试者加入了公司，双方会不会“和睦相处”（面试者升职加薪，公司project完成得好，业绩越来越好），而不是面试者performance不行/不喜欢公司没多久就跳槽等，导致工作没法完成浪费资源

而这些答案，来自于面试过程中的每一个细节，来自于双方的不断的语言/非语言沟通，来自双方的面试体验和感受，因此，面试重在过程

找到一个理想的公司拿到好offer对于我们很难，但对于公司来说，找到一个理想的合适的员工也很不容易，毕竟面试也是要花很多时间成本及资金成本的(想想你的机票酒店啥的)。这也是为什么面试有那么多轮，要和那么多面试官聊天，甚至必须当面聊天，必须现场体验公司真正的环境，要问我们那么多问题，也要我们问他们那么多问题

与之相反，考试/刷leetcode，是一个静态的，单方面的过程，只关注一个solution是accepted/wrong answer，重在结果而忽视了过程。楼主个人的经验是，“题做出来了与否”，**不能作为单一的因素判断面试是否能通过**，有可能就算你的solution，或者你觉得你的solution是对的，但面试官不喜欢跟你一起解决问题的过程，觉得不爽，那面试结果也未必会好。

更为重要的是，能否通过面试，最后能否拿到offer这个结果，可能是一件很复杂有很多很多不确定因素的事情：

- 面试的题目难不难，自己接触过类似的算法没有
- 面试官会不会给你放水(有的国人小姐姐甚至会用中文面试)
- 每一轮面试的表现/面试官对你的感觉
- 你的tech/style是不是很match这家公司/这个组
- 你现在HC多不多/缺不缺人，今年很多公司因为HC告急导致面试bar升高([airbnb](#)/GG等)
- 你在waiting list的排位在哪里 (Vmware/[Amazon](#)根据waiting list发offer的)
- 再往大了说，经济下行/肺炎/政策这些完全随机的事件，都可能对公司收益有巨大影响，从而影响到HC和offer结果
-

总之，运气对于面试结果，真的很重要，但也真的完全不受我们控制，我们只能把自己能做好的过程做好，提升自己的硬实力软实力，看淡结果，注重并总结过程中的得失，以不变应万变，并保持积极乐观，要相信：越努力，越幸运。

面试时的一些小tips

- 面试把过程做好就够了，不用太在意结果，把面试官当作同学同事一样一起讨论解决问题
- 沟通就是展示自己的idea，同时接受别人的feedback，所以如果面试官的feedback/表情有点不好，也不用紧张担心，接受它并看下一步怎么做，事情未必有我们想象的那样糟糕
- 遇到问题了多问，ask question也是工作中重要的能力，别人怎么看你都不重要，make things done/解决问题才重要
- 过程中遇到分歧了，或者遇到了各种突发情况，听面试官的(也是take feedback)，跟着面试官走好，不要头铁不要头铁不要头铁
- 实在是紧张焦虑让自己不能好好思考时，可以告诉面试官自己有些紧张，大部分人都可以理解，然后双方微微一笑，气氛就会轻松很多
- **面试最好的状态，就是心无旁骛，一心投入并享受跟面试官一起讨论问题解决问题的这个过程，be yourself ~ 面试结束后，关注点也是跟面试官的合作是否愉快，如果自己有不会的地方，有没有从中学到新的东西 ~**

面试沟通模版(针对onsite算法面试，phone同理)

1. 面试官描述题目，onsite的题目描述基本不会像phone那样直接粘贴描述，而是：
 - a. 说要解决的问题的back ground
 - b. 给一个base的test case
 - c. 然后说问题，并给出base test case的答案
 - d. 一般不会主动告诉input和output的format
 - e. 一般而言，onsite的问题描述都是比较抽象的(为了模拟实际的问题环境)，哪怕是lc的题，也不会像lc或者phone那样给很多的detailed information，需要candidate自己去ask question来clarify.
2. 面试者(你)问clarification的问题
 - a. 问清楚要解决的问题具体是什么，什么input，什么output等，并征求面试官的确认
 - b. 问清楚了题目中的所有可见的或者隐藏的condition/constrain，并征求面试官的确认
 - c. 如果自己有意识/无意识地make assumption，像面试官确认这些assumptions
3. 面试者(你)问test case/corner case
 - a. 一般面试官会和你一起讨论你提出的testcase，看正确答案应该是怎样
 - b. 一般在这里如果test case没有想全，尤其是cornercase，那么之后在想idea并walk through的时候一般会出问题
 - c. 一般而言，面试中，在开始想idea以及walkthrough idea之前，先跟面试官把test case/corner case讨论全，会对之后的idea思考大有帮助，很多时候idea不work就是有些case根本不能handle导致的。
 - d. **小技巧：对于某些面试题，或者面试官，部分困难的test case甚至可以不求实现，或者算作follow up，但candidate要能够想到这些case，会是一个加分点**
4. 面试者(你)告诉面试官，能否让自己想一会儿idea(1 min左右)，然后跟他一起讨论想到的idea：
 - a. 因为时间紧迫，不能花好长时间站着不说话干想，建议先思考high level idea
 - b. 如果high level idea跟正确的idea方向一致，可以进行下一步，讨论idea的细节并walkthrough->面试官快速+满意的笑了/点头
 - c. 如果high level idea跟正确的不一致，或者是恰好错误/不够好的方法，可以先跟面试官一起分析一下题目，看看哪里是不够好，行不通->面试官快速地反馈给feedback
 - d. 如果某个点被block了，向面试官ask for help，告诉他在哪个地方confused了，好的面试官都会带着你往正确的track上走
 - e. 如果自己的idea面试官不确定行不行，先进行下一步->面试官慢速地思考/犹豫/疑惑/提出问题
 - f. 一般而言，跟面试官讨论idea的过程中，可以通过对方的表情/神态判断出自己的idea是什么状态(恰好正确/完全错误/面试官没见过不确定对错)

- g. **小技巧: 先提出一个brute force的idea并分析复杂度, 可能就是一个可以写代码的解, 或者缓解想不出来的尴尬气氛, 不至于什么idea都没有**
5. 面试官(你)讨论idea的细节, 用什么data structure, 并用test case来walk through这个idea
- 如果自己的idea错误/不够好, 面试官一般会给一个testcase来attack这个idea, 但能自己想出来更好
 - 如果自己的idea的细节有block, 也可以向面试官ask for help一起讨论
 - Ask for help本身就是工作中重要的一种能力, 也是面试中要评测的能力, 寻求帮助一点错都没有, 不丢脸**
6. **讨论/讲解idea的每一步, 都要确定面试官对自己的idea和行为完全ok, 当然, 更重要的是, 自己也要对自己的idea完全清晰的理解, 再开始写代码**
- 白板写代码, 要注意排版和字体间距, 尽可能让面试官看得舒服, 很多面试是要求面试官拍照回去自己敲一遍
 - 每写一部分, 可以回过头跟面试官说一下这个是干什么的, 并确认面试官理解了
 - 小技巧: 把一个完整的算法代码拆分成几个函数, 其中一定要把main logic这个部分独立出来, 然后先把high level的structure和各个helper function的接口定义好, 然后着重实现main logic的部分, 当然, 记得征求面试官同意, 不然他不知道你在干什么**
 - 一般而言, 在算法面试中, 一定有一个面试官最希望看到的部分, 就是算法的main logic, 比如dp的状态转移方程, 只要这个部分写好了, 这道题的solution就成功了一大半, 因此要把主要时间留在这里而不是其他的边角余料, 比如data structure的初始化这种完全可以放在helper function里面, 这些东西哪怕不写在很多情况下也是ok的, 前提是征求面试官同意
7. Walk through test case(如果前面已经做了, 这一步可以省略), **test case的walkthrough一定要有, 不管是在写代码前的idea讨论, 还是写代码后的debug, 根据面试官的要求来(要问)**
8. 主动对代码的时间复杂度/空间复杂度的分析, 可以在代码之后也可以在ideawalk through部分, **一定要自己主动提出分析复杂度, 体现自己算法analysis能力, 面试官会期待candidate自己主动来分析算法**
9. **小技巧: 楼主之前onsite一直因为心态不好崩掉, 后来把面试模版总结了一下写在了小纸条上, 每次面试前都会看一下, 心态就会好了很多, 因为每一步要做什么都很清楚了, 就不会紧张**

算法/数据结构复习(刷题)

- 刷题法题不在多, 而在精, 每做一道题最好能够理解透彻, 不是只记住了solution就开始下一道, 而是多思考, 并清晰的理解solution如何得出来的这个过程
- 刷题法题优先按照算法数据结构或者解题类型tag把经典的一些题复习一遍, 不用做特别多但要知道每种数据结构/算法/解题方法怎么用, 什么时候可以用, 为什么可以, 复杂度都是多少, 这样避免面试中考到了完全没见过的算法/数据结构而完全懵逼
- 其次优先刷经典且高频的题(比如LC前200, LRU, trapping rain, countingislands等等), 这些题因为很经典, 楼主发现在现在的很多面试中依然在考或者考他们变种, 哪怕他们题号未必很高
- 再次可以刷刷公司tag, 面经, 以及lc contest, 楼主觉得打lc contest可以比较好的通过rating反应自身算法题思维能力的成长
- 楼主自己在之前刷题准备过程中犯过的一些小误区:
 - 一定要把公司tag都刷完才能去面试吗? 答: No, 不能完全依赖tag题库/面经题库去面试, 楼主面BB的题都不在BBtag和面经里, 面MS的题甚至都不在lc里。。。面试最重要的还是看思维能力+沟通/面试技巧, 多刷10道题不如多mock一次提升一下沟通的性价比
 - 马上就要面试了时间不够了耶, 可以背题目的solution吗? 答: No No No!!! 楼主后来发现, 面试重要的是过程而不是结果, 面试官相比于你的solution更看重你思考问题和沟通的过程, 而背了一个不理解(solution, 楼主发现展现思维过程就变得很不自然, 因为要从结果倒推原因, 很容易导致面试沟通紧张。更重要的是, 好几次面试楼主一开始没有想出来但跟面试官要hint沟通最后做出来, 也能通过。而很多人面试中体会过的, "我题都写出来了, 怎么还是挂了?", 不也说明, solution正确与否, 未必有沟通, 和让面试官爽到重要吗?
- 尽早准备白板写代码, 多练练白板写代码的排版/字迹啥的, 让面试官看着舒服

面经

- 跟刷题的第5条所说的类似, 不能完全依赖面经, 而是把面经当作没做过的新题, 完全理解地去刷, 因为很多公司(GG/BB/MS)题库大的吓死人, 还会定期更新题库, 地里面经虽然很多, 但全部都准备工作量太大了, 还很有可能面试一道题都没见过, 所以性价比不如多mock提升沟通和面试技巧(楼主身边过了GG的好几个都是裸面, 而也有几个把GG面经tag刷了很多的intern conversion面试都是没见过的题然后跪了), 楼主自己BB/MS/GG都不是面经题, 或者是面经题但是面试前实在是没有时间把它们都好好准备到, 最后GG还是挂在沟通上了
- 很多题库小的公司, 而且面经非常固定的公司(2sigma/Akuna/Lyft/Airbnb/Jane street。。。), 还是可以好好刷面经的, 性价比还是很高的
- 而且, 对于一些面经少的可怜的公司(Citadel/HRT。。。), 看面经和裸面没啥区别啊
- Debug的题目面经慎看, 只要看了答案, 面试找bug就全靠演技了, 而楼主演技不好, 所以面试时这一轮感觉不如不看面经
- 最后, 楼主觉得, 面经最大的好处, 就是, 让你了解公司喜欢考什么样类型的题/知识, 然后就可以针对地系统地复习这些知识, 比如金融fintech公司喜欢考C++和OOD, 有些硬核公司比如VMware喜欢考OS和Network, Waymo做planning的组喜欢考geometry, palantir热衷BQ等等

Behavior question(BQ)

BQ推荐参考一下地里两个BQ神贴, 楼主当时找工作也是看了这两篇帖子后BQ渐渐好起来, 感觉非常有用:

<https://www.1point3acres.com/bbs/thread-566677-1-1.html>

<https://www.1point3acres.com/bbs/thread-106072-1-1.html>

为什么面试要考BQ? 因为technical面试, 不管是算法题还是system design还是各种技术问题, 其实都是为了回答面试要解决的第一个问题: 面试者的技术背景和解决问题以及沟通能力是否满足公司的要求。而BQ是为了回答后面三个问题, 楼主个人的面试体验是, 绝大部分BQ问题都逃不出这3大类: (以下例子都是楼主在面试中实际遇到过的, 自己觉得比较典型的例子)

- 面试者在工作上是不是足够professional
 - Examples:
 - 你在项目中遇到了什么challenge->你有没有make things done的能力
 - 你有没有任务ddl非常紧的时候->你有没有时间管理的能力
 - 你和同事有conflict了怎么办, 有没有persuade/be persuaded的经历->你有没有communication/collaboration的能力
 - 。。。。
 - 总的来说, 准备这些问题时, 可以从这样两个步骤入手:

- i. 提前了解工作中有哪些behavior是professional的哪些不是, 可以是实习/工作中领悟的, 也可以是帖子/网课里看的, 这里强烈推荐一下地里K姐的职场晋升, 职场新人, 职场情商等网课, 对于没有很多工作经历的人真的挺有用的
 - ii. 结合自己的项目经历, 总结一些小故事, 这些故事要能够体现这些professional的能力, 不一定每个能力一个故事, 楼主准备的故事里往往一个故事可以对应多个点, 然后发现面BQ时流程快了很多, 因为很多面试官想问的点你已经一次回答出来了, 而且非常自然
 - c. 最后, 建议大家能够深刻理解这一类的BQ问题, 楼主认为, 这样不仅是对通过面试有用, 对于工作本身更是有用。楼主就是一开始不懂这些东西, 自以为是地把实习return玩崩了, 一开始的好几个BQ面试也都崩了后才开始痛定思痛深刻反省。很多时候工作和面试不应该分开来看。。
2. 面试者是不是真心喜欢公司, 认同公司(文化/技术/氛围/mission等)
- a. Examples:
 - i. 为什么选择我们公司? ->你对我们公司有多了解? 花了多少时间research我们公司? 了解公司的文化/mission吗?
 - ii. 你有什么问题要问我们的吗? ->毕竟你现在对公司还没有特别了解吧, 如果你真的想来, 一定会有很多东西想了解的
 - b. 总的来说, 有这么几点可以入手:
 - i. 做research, 提前了解这个公司的文化/mission
 - ii. 夸!!! 但要基于事实的夸, 不要胡编乱造, 不然多尴尬。不过夸一番肯定会让大多数面试官很开心, 面试官一爽, 给你的feedback自然也会好一些啊哈哈
 - iii. 小技巧: 准备问面试官的问题前, 把自己当作是已经收到了他家offer, 决定要不要入这家公司的新员工那样, 这样你自然就会有很多问题, 毕竟是真正地要参与工作而不是为了过面试, 这样自然就会有很多担心和顾虑, 问问题也自然很多。(据个人经验, 大家在比较offer时间HR和公司内部员工的问题, 是面试中的好几倍(捂脸))
 - c. 最后, 面试是双向的不是吗? 这个问题本身也是一个机会, 让你真正了解这家公司, 了解这家公司跟自己的兴趣是否吻合, 了解自己是不是真的喜欢这些员工, 并向往在这里工作, 同时这也是对自己的职业发展的负责。
3. 面试者的职业兴趣和职业规划是否与公司, 以及这个对应的职位相match
- a. Examples:
 - i. 介绍一个你最喜欢的项目->你喜欢什么样的project/work/impact->你会喜欢我们公司的project/work吗?
 - ii. 你是master? 你在学校有什么研究方向吗? 或者偏好什么课吗? ->同上
 - iii. 你的简历当中的经历有些多样化, 你之后更想从事哪一个方向呢? ->你的职业发展期望是什么?
 - iv. 你如果入职了, 我们这里有***这些组, 你有想去的组或者想做的东西吗? ->这个就要看你对公司的research做的怎么样了
 - v.
 - b. 总的来说, 可以注意这么几点:
 - i. 这类问题常常出现在你自我介绍结束后, 面试官看完你简历后, 根据你的简历来问, 所以投每家公司的简历尽量差异化针对化, 这样不至于让面试官看了一圈简历project发现跟公司project/work一点关系都没有就很尴尬
 - ii. 还是要提前做好research, 了解一下公司具体的work/project有哪些, 跟自己的经历里有哪些可以match的点, 这样聊的时候就比较容易说, 而不显的生硬尴尬。
 - iii. Example: 楼主自己的例子是, 去面了一家fintech的公司, 但自己其实对金融一窍不通, 但自己有很多C++的经验, 所以问我这些问题时, 我就说自己是C++ fan, 但科技公司里backend用C++的不多, 所以对你们家做trading system相关的工作比较感兴趣, 尤其想试试lowlatency相关的组blabla

System Design

因为楼主面试的system design非常少, 所以就不给经验了。-当然, 可以肯定的是, 跟算法面一样, 系统设计面也很看重沟通, 沟通, 沟通。不过值得一提的是, 有些面试中会夹杂着考一些system的知识比如concurrency, transaction, multi-threading这些, 所以转专业的同学们有精力选一个OS, 或者了解一下应该对面试也会有帮助的。

面试规划+个人向推荐timeline

1. 实习要认真做好, 按照K姐的职场新人课讲的来做, 争取拿到return offer, 有return找工作是easy难度, 没有return的找工作, 是地狱难度(心里压力会非常非常大)
2. 7月末8月初就有很多公司开放职位, 要开始准备好简历开始投了, 这个时候是拿面试最容易的时候, 一旦到9~10月海量求职者很容易填满面试的坑位, 到时候机会就少很多了
 - a. 这个时候刷题/design/BQ都还没有准备好怎么办? 答: 不慌, 投出简历到收到面试差不多需要1~3周左右的时间, 拿到面试后也可以再往后约几周, 时间是够的
3. 8月刷题+准备BQ, 8月末9月初可以开始mock, 而且尽早准备mock白板, 尽可能在早起把基础都打好, 因为9~10月面试高峰期非常辛苦, 可能还有考试和作业什么的, 很难有大量时间再准备这些了
4. 面试安排:
 - a. 找工初期以上岸为首要目标! 找工初期以上岸为首要目标! 找工初期以上岸为首要目标! 不管多小的公司, 是不是tech, 只要能SDE岗位sponsorOPT/H1B的都可以去试试, 拿到一个offer后心态会好很多很多, 多面小公司也可以积累面试经验, 有offerddl后可以大幅度加快后面的面试进度(催面试/催结果。。。), 而且小公司面经都比较少, 容易准备。
 - b. 9月会陆续收到一些phone screen, 一般来说phone的bar相比onsite没有很高, 而且纯算法题居多, 很多公司甚至就没有phone, 只有OA或者简单的karat, 所以准备压力不会很大。前期差不多1周1~2次面试, 每次面试完后要花时间总结一下takeaways, 看看是哪里不足(思路/沟通/面试技巧。。。), 并加之改进, 再开始下一次面试。
 - c. 不要把面试拖晚了! 不要把面试拖晚了! 不要把面试拖晚了! 很多公司今年HC很少, 如果拖晚了很可能就没有HC了面都面不了(比如Airbnb, 10月初就结束招聘了, GG也结束的比往年早), 不需要把题库/tag/面经都刷完准备得万无一失才能去面试! 实力+沟通+面试技巧才是最重要的! 尽力去面就好了, 面不面的过有时也很看运气, 说不定HC还充足就过了呢, 相反多刷了2周题HC可能就没了, 而且准备地越久效率越低, 压力也越大。
 - d. 9月和10月开始就会有onsite, onsite也是可以一周只面1家, 因为飞来飞去非常消耗精力, 每次面完后及时总结feedback并改进提升自己, 把每次面试当作是学习的机会, 最好还要和几个小伙伴一起找工作, 可以互相分享信息交流面试经验, 以及面试不顺的时候互相安慰。。。
 - e. 拿到第一个比较理想的offer后, 就可以把同一个城市的onsite集中在一起几天内面完, 这样面起来轻松很多, 而且基本到这个时候面试能力/算法能力/BQ基本都收敛了。如果可以尽量不要1周内去东西多个城市onsite, 累到吐血(楼主有一周连着onsite了西雅图, 纽约, 湾区, 浪费了大量时间在飞机和路上, 还很疲惫)
5. 总之, 面试不要贪求一步到位进大厂, 有的时候一步一步来反而更轻松, 楼主身边包括楼主在内, 在没有offer之前都一直跪, 从拿到第一个offer开始, 心态好了自信来了, 之后的面试都顺利了很多。

最后, 找工作, 真的就是努力再加上那一点点的运气, 希望这篇文章中的技巧和经验, 能帮助到正在找工路上和即将在找工路上奋斗的同学们, 也希望大家的运气能尽早到来, 给大家带来理想的

offer ~

PS: 因为个人写的总结, 所以很多内容只是根据个人的找工经验有所侧重, 比如刷题和前期项目准备, networking都没有写很细致, 精华内容都放在面试这一块。推荐一下这两篇帖子, 很多地方总结的比楼主要好:




<https://www.1point3acres.com/bbs/thread-532585-1-1.html>, <https://www.1point3acres.com/bbs/thread-594973-1-1.html>

本主题由 admin 于 4 天前 提升

评分

参与人数 **78** 大米 **+350** 理由

收起

 李冬冬	+ 2	给你点个赞!
 Harry·Sun	+ 1	很有用的信息!
 zhen2019g	+ 1	很有用的信息!
 white1997	+ 1	赞一个
 xlin05	+ 1	给你点个赞!
 好吃的芝士蛋	+ 5	给你点个赞!
 andychen96	+ 1	很有用的信息!
 Sweetbear	+ 2	给你点个赞!
 Zampano	+ 1	给你点个赞!
 littlebambib	+ 2	给你点个赞!

查看全部评分

上一篇: 担心verbal offer要被撤回 求助

下一篇: 国内字节跳动DA sp 与湾区juniper network SDE 如何选择

本帖被以下淘专辑推荐:

· [DS/DA 找工求职经验](#) | 主题: 103, 订阅: 85

· [学习策略与预期目标](#) | 主题: 9, 订阅: 0

· [SWE找工作](#) | 主题: 2, 订阅: 0

★ 收藏 338 🍌 评分 ❤️ 分享 2 📁 淘帖 3



3M Particulate Respirator 8210, N95, Smoke, Dust, Grinding, Sanding, Sawi...
\$279.00 ~~\$349.99~~

(128)



Earloop Face Mask, Sanitary Surgical Face Shields, Dust Mask, Blue (250)



Nishiki Medium Grain Rice, 80 Ounce

\$6.98

(452)



Braun Digital Ear Thermometer ThermoScan 5 IRT6500

\$78.19

(4131)

回复 微信

使用道具 举报



我的人缘 0

9 主题 | 67 帖子 | 79 积分

发消

CHEWBUEW 3 天前 | 只看该作者

推荐

本楼: 92% (36) 7% (3)
全局: 92% (454) 7% (39)

本帖最后由 CHEWBUEW 于 2020-3-6 12:33 编辑

作为一个已经拿到FAANG offer的人，表示不赞同找工作是努力加一点点运气，而是90%运气+一点点努力，还有刷题那块数量一定要多，700是底线，你说在于精不在多，怎么算精？你只刷前300能接触到fenwick tree/targan/三维dp/Max Flow/匈牙利算法/背包问题的九种套路吗？当然不可能接触到，你肯定又会说这些不会考，我告诉你我都被问到了，而且不止一家公司，你没遇到只是运气好而已，这就是90%运气+一点点努力，有人面FB就是两道2 easy，有人就是45分钟burst balloon + sky line，你找谁说理去？我现在已经不相信刷够xx，或者周赛进xx就能去xx公司了，或者xx公司没问题，yy公司肯定能过的言论了，话我就放在这儿，进FAANG的不是因为比你牛逼，就是因为运气好或者你是个女的罢了，没上岸或者再ICC里的刷题1000+，周赛每次前100的比你强的多了去了，你只是在漫无边际的题海里抽中了你恰好会做的题而已，让Google里的人再面试一次，有几个人敢打包票自己一定能过的？

○ 评分

参与人数 7 大米 +9 理由

收起