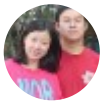


Only you can see this message



This story's distribution setting is off. [Learn more](#)

和大家分享我七天Onsite面试硅谷七家公司的经历



hua zhang

Oct 21, 2018 · 7 min read

大道至简

尘世中迷途小码农

我的上一篇文章： ME PHD quit, 转CS 找工作经历

今天（2018-10-10）我的H1B transfer 终于被 approved 了，心中的大石终于放下来了，可以和大家分享一下我找工作的经历，也是对我自己的一个总结。

H1B timeline:

08/29 LCA filed

09/07 H1B filed

09/19 RFE received (Major does not match)

09/28 RFE doc filed (感谢律师这么努力及时的给我准备资料)

10/10 Approved

由于PTO有限，所以我决定将所有公司的面试安排在同一个时间，集中突破。拿到的offer有：Airbnb, Google, Uber, Opendoor, Axon, Dialpad。由于Palantir出结果太慢了，我就直接给HR说已经去了别家。Thumbtack的话，由于实在太累了，就把onsite给推掉了。

准备阶段：

1, 算法题目的准备

leetcode的题目多到令人可怕（913道），由于白天经常加班，最后陆陆续续的刷了大概90多道题目（Easy 13, Medium 52, Hard 29），前后大概花了3个月。我刷题是按照分类来集中突破的：

- Graph: 平时上班基本上没有用到Graph，也是面试中很非常常见。强烈推荐hackerearth 上面的文章：链接。最好能把里面每一个算法都看一遍，写一遍。
- DP: 面试高频。多做leetcode相关题目，总结做题规律
- Sort: 各种sort的算法，最好自己能动手把每一个算法都写一遍
- Disjoint Set Union: 我也被问到过好多次。这篇文章非常好：Disjoint Set Union (Union Find) — Prateek Garg
- Trie Tree: 很有必要掌握。Trie (Keyword Tree) Tutorials & Notes | Data Structures | HackerEarth
- 其他 刷题的感受就是，自己写完之后，即使答案被accept之后，也一定要：*
多想想可能会有什么样的test cases，甚至一些变体。比如一些DP的题目，可能会让你找一个最小或者最大的值，这个时候你可以想想，有没有办法找到那个路径呢？我也有经常面试别人，我有时候就会这么问求职者。
- 多看看别人的算法和代码，看看这么优化。有时候你会看到令人虎躯一震的解法。
- 个人感觉用high level的语言写的时候会比较快，比较有优势，尤其是一些要让你跑test cases的公司。

2, 系统设计

个人感觉系统设计的准备不仅对面试有帮助，而且在工作中也会起到很大的帮助，所以希望大家认真准备。我准备的方法是先看看网上的一些例子，强烈推荐：Grokking the System Design Interview，里面的例子非常的好。面试官可以问的系统设计非常的广泛，而且可能每一次focus的细节也不一样，所以上面的例子只能给我提供一个框架。大概步骤：

- • 先明白需求，和面试官讨论需求
- • 潜在用户
- • 提出一些可能的应用场景

- • 系统限制
- • 估算系统可能的traffic, traffic per second。还可以谈及peak traffic, traffic micro burst。这里要学会怎么估算。
- • 100M requests per day \rightarrow \sim 1100 requests per second
- • 最好可以调查一些大公司的数据, 比如说 daily active users, total user base, number of tweets per day, number of queries that Google handles per day. 这样在估计的时候不至于太离谱。
- • 估算数据库的大小, 1年后, 5年后, 10年后。这里要知道每一个data type的大小, 已经估算可能存储数据的大小。
- • 估算硬盘的大小, 1年后, 5年后, 10年后。
- • High-level 的一些components, 以及他们的interface
- • 知道系统的限制在哪里, 提出可能的解法方法: *
- • 记住没有最好的方案, 重点要讨论你要保证什么, 牺牲什么 3, 简历项目准备 至少准备一两个简历上面的项目:
- • 你项目哪里可以优化如果你重新做的话?
- • 你的项目最困难的地方在哪里? 怎么解决的 面试 1, 投简历 真诚感谢那些帮我内推的同胞们, 感谢他们的大力内推, 我才能拿到那么多面试。

2, 电面

就一个感觉电面的要求越来越高了, 很多公司都要求现场跑测试。。。我把所有电面都安排在了我的中午, 中午出去一个小时, 就当出去吃午饭。Airbnb: 电面两次

Google: 拿到Airbnb onsite后, Google直接给onsite了

Uber: 一轮电面, 问了一个很有意思的图的问题

Opendoor: 一轮电面后, 说要加面, 理由是我知道电面问题的一个概念, 然后面试的内容和概念基本无关。。。我就和Recruiter商量直接onsite, 说我已经在那一周安排了好几个面试, 我可以直接去他们, 她还是坚持要加面。我就说你可不可以和你manager商量汇报一下, 如果不行的话, 就加面。最后就直接onsite啦。还是得和他们多扯皮。

3, Onsite

第一天：Google（周四）

作为每一个码农都向往的地方，自然是要试一试了。由于之前skip了电面，onsite有五轮面试，每一轮都是白板。感觉他们更多的是考察你够不够聪明，能不能把复杂问题简单话。考察了比较多动态规划，还有图。

第二天：Airbnb（周五）

感觉bar很高，公司也很大方，给我cover了三天的住宿 :) 两轮coding：现场要跑test，跑不出来基本上是挂。我第二轮code没有写好，可能是午饭吃多了？？一道很简单的题目没有写出来，等面试官走出去后，立马想到怎么写，赶紧冲出去给update。。最后要求加面 两轮system design：也不难，主要是要和面试官多讨论，说出你的想法，就好想你们在一起设计这个系统，一起工作。两轮culture fit：特别喜欢他们家的culture fit的面试，可以感觉到他们很用心的筛选面试者，以保证可以很大程度的match。很喜欢他们的办公室：会议室的布置很多都是根据比较好的host的风格来布置的。BTW：公司可以带狗上班。面完之后就已经收到Google的offer了，狗家的速度真心快

第三天：Dialpad（周一）

一个比较早期的startup，自己也用过他们的产品，印象挺不错的，所以就试了试。整个面试都不难，题目基本上很快就可以写出来。

第四天：Uber（周二）

两轮code：一轮code比较简单，没有什么特别的地方。第二轮特别有意思，先给你一个很抽象的问题，然后让你慢慢把问题抽象成一个数学问题，然后变成一个计算机代码问题。两轮系统设计：多讨论，自己不太懂的名词就尽量避免，因为面试官很有可能就会一直问下次，那就比较悲剧了。一轮bar raiser：问问之前的工作经验。一轮hiring manager：问一个很简单很简单的代码问题，然后就聊聊简历上面的项目

第五天：Opendoor（周三）

面试过程比较新颖，和面试官一起pair写代码。你要让他们注意到你的代码风格，比如哪里要注意什么corner cases，unit tests怎么写等等。有一轮聊你做过的项目，需要说的很详细，说你当时为什么这么选择，好在哪里？不好在哪里？第六天：Palantir（周四）有一轮面试非常有意思，面试官直接把电脑拿出来，介绍里面的代码，然后说两面有两个bugs，看你是怎么一步一步找到bug，然后看你怎么修。其他轮都比较中规中

矩，白板写代码，系统设计 BTW：公司可以带狗上班。第七天：Axon（周五）强烈推荐的一家公司。公司主要是给警察做产品（最开始做taser的），做软件系统，市场非常大。我面的是一个比较新的部门，Axon Records，有很多的机会。还可以和警察一起执勤。他们办公室的大门 面试题目还挺难的，两轮代码题目，其中一轮就是他们现实中遇到的一个问题，面试官给你描述问题背景，然后你抽象出数学问题和数据结构，然后给出解法。系统设计题目也是以他们现在的系统为背景，让你设计以及优化。在机场上收到了Uber的offer。管理：我用Trello来管理我申请的公司，弄了不同的list来管理每一个公司的状态。感谢贝乐影视 — 在线免费高清电影！让我在工作刷题之余放松一下自己。

Yosoro

[About](#) [Help](#) [Legal](#)