# 2021三月月报



## 技术相关

### 字节跳动Data数据中台后端实习

#### - 面试复盘

- 2021.02.22 官网投递简历（内推）
- 2021.02.23收到HR电话通知面试，一面安排在02.28，为自己留足时间作准备
- 02.24～02.26，准备面试，快速回顾了剑指Offer所有题目，计算机网络、数据库面经过一遍，学到了不少新知识
- 面试前两天，实际上感觉已经没什么可准备的了
- 2021.02.28 16:00，一面，面试官是一个小姐姐，先自我介绍，然后考察了python基础、计算机网络和数据库，最后做了一道算法题（链表奇数节点单调增，偶数节点单调减，要求空间复杂度O(1)的情况下把链表排成有序），第一次面试有点紧张又有点着急，做得很差，花了很多时间，好在面试官特别照顾，直到我做出来才宣布面试结束，连反问的环节都跳过了，然后跟我说马上就可以二面
- 2021.02.28 17:30，二面，同样自我介绍开头，聊了聊前端营的项目，然后考察数据库以及一些零散问题（我至少有连续4个问题回答我不知道.....），但事实证明其实这些问题不

会对最终结果的影响很小，只要代码能敲出来，其他的都好说。最后做了两个算法题（顺时针打印矩阵，和找出无序列表中第k大的数），都是剑指Offer原题，直接秒，二面结束。五分钟以后收到HR小姐姐电话约第二天下午三面

- 2021.03.01 15:00，三面（原定14:00开始，推迟了一个小时），自我介绍，聊了聊暑期做的CIFAR神经网络入门项目，问了问我做的过程，问我主页里这么多项目都是自学的吗？然后说看我都做了这么多题，这一面就不做题了（卧槽还有这种好事？？），于是让我自己挑两个我刷题过程中印象比较深刻的题给他讲一讲.....最后给我讲了一下数据平台大致的情况，总而言之三面没有一个技术问题，就是简单聊天，20分钟就结束了。五分钟以后HR电话约马上四面
- 2021.03.01 15:30，四面，HR面，也是聊天，在此不细说了，跟我说Offer需要两到三天审批
- 2021.03.02 短信联系了HR小姐姐询问面试情况，小姐姐说通过了四面，还在审Offer
- 2021.03.03 16:00，HR加了我微信，确定入职时间后发了offer！

## - 三月实习工作

- 本着遵守内部信息安全原则，不应在内网以外发布有关企业及部门的工作相关信息，因此我仅介绍入职以来的一些体会与感受
- 字节工作的整体节奏非常快，我作为日常实习生，在几乎没有任何后端工作经验的情况下，在入职的第三天就开始要处理一些简单的需求，入职一周就要把更新的部分推上线，这对我而言是很大的考验。刚刚接触业务，既怕效率低，又怕出错误.....
- 字节内部的办公系统非常好用，食堂伙食相当不错～
- 包括实习生在内的每个字节员工都要在处理工作的同时参与到招聘工作当中来，有几年工作经验的员工也会直接参与面试环节（字节有很大一部分员工会参与到面试工作）
- Golang正在被越来越多的企业运用，学好Golang，既把握了现在，又投资了未来～
- ByteStyle 字节范
  - 开放谦逊
  - 坦诚清晰
  - 始终创业
  - 务实敢为
  - 追求极致
  - 多元兼容

- 刚刚入职，还不太适应，工作特别多，本月所获置于四月月报中

---

## .算法总结

- 动态规划 -- 卖股票

- 121. 买卖股票的最佳时机

```
# 只买入一次，类单调栈问题
class Solution:
    def maxProfit(self, prices):
        value, res = prices[0], 0
        for i in range(1, len(prices)):
            if prices[i] > value:
                res = max(res, prices[i] - value)
            else:
                value = prices[i]
        return res
```

- 122. 买卖股票的最佳时机 II

```
# 可以多次买入但不能同时参与多个交易（最多持有一张股票）
# 动态规划思路
# 每一天结束时只会存在两种状态：
# 1. 当前不持有股票
# 2. 当前持有股票
# 考虑动态规划的递推关系：
# 1. 当天不持有股票有两种可能：前一天不持有股票且当天不买入；或者前一天持有股票但当天卖出
# 2. 当天持有股票也有两种可能：前一天持有股票且当天不卖出；或者前一天不持有股票但当天买入
class Solution:
    def maxProfit(self, prices):
        dp = [[0, 0] for i in range(len(prices))]
        dp[0][1] = -prices[0] # 定义初始状态
        for i in range(1, len(prices)):
```

```
            dp[i][0] = max(dp[i-1][0], dp[i-1][1] + prices[i]) # 当天不持股状态
            dp[i][1] = max(dp[i-1][1], dp[i-1][0] - prices[i]) # 当天持股状态
        return dp[-1][0] # 返回最后一天不持股状态
```

- 309. 最佳买卖股票时机含冷冻期

```
# 冷冻期指的是卖出后一天内不允许买入
# 动态规划思路
# 每一天结束时会存在三种状态：
# 1. 不持有股票，且当日没有卖出
# 2. 持股
# 3. 不持有股票且当日卖出
# 考虑动态规划递推关系：
# 1. 当天不持有股票且没卖出有两种可能：前一天不持有股票（包括前一天不持有且没卖出以及前一天卖出）
# 2. 持股有两种可能：前一天就持股；或者前一天不持股且没卖出，当日买入
# 3. 不持股且当日卖出只有一种可能：前一天持股，当日卖出
class Solution:
    def maxProfit(self, prices):
        dp = [[0, 0, 0] for i in range(len(prices))]
        dp[0][1] = -prices[0] # 定义初始状态（如果持股，买入需要花费金额）
        for i in range(1, len(prices)):
            dp[i][0] = max(dp[i-1][0], dp[i-1][2]) # 不持股且当日没卖出
            dp[i][1] = max(dp[i-1][0]-prices[i], dp[i-1][1]) # 持股
            dp[i][2] = dp[i-1][1] + prices[i] # 不持股且当日卖出
        return max(dp[-1][0], dp[-1][2])
```

- 123. 买卖股票的最佳时机 III

```
# 最多可以完成两笔交易，且最多持一股（手中无股票才能卖入）
# 动态规划思路
# 每一天结束时存在5种状态：
# 1. 没有任何操作
# 2. 第一次买入状态
```

```
    # 3. 第一次卖出状态
    # 4. 第二次买入状态
    # 5. 第二次卖出状态
    # 考虑动态规划递推关系:
    # 1. 没有任何操作，说明之前也没有进行任何买入（实际就是保持为0）
    # 2. 第一次买入状态，有两种情况：前一天没有任何操作，当日买入；或者前一天已经是第一次买入状态
    # 3. 第一次卖出状态，有两种情况：前一天第一次买入状态，当日卖出；或者前一天已经是第一次卖出状态
    # 4. 第二次买入状态和卖出状态同上，略
    # 返回一定是卖出状态（或者没有任何操作），手中肯定不持股
    class Solution:
        def maxProfit(self, prices):
            dp = [[0, 0, 0, 0, 0] for i in range(len(prices))]
            dp[0][1], dp[0][3] = -prices[0], -prices[0]
            for i in range(1, len(prices)):
                dp[i][0] = dp[i-1][0]
                dp[i][1] = max(dp[i-1][1], dp[i-1][0] - prices[i])
                dp[i][2] = max(dp[i-1][2], dp[i-1][1] + prices[i])
                dp[i][3] = max(dp[i-1][3], dp[i-1][2] - prices[i])
                dp[i][4] = max(dp[i-1][4], dp[i-1][3] + prices[i])
            return max(dp[-1][0], dp[-1][2], dp[-1][4])
```

# 反思与认识

## 关于当下和未来

- 大三暑期看各个大厂入职面试的题目，感觉离自己特别遥远，觉得凭大学三年学的知识完全无法让我游刃有余得找到哪怕一份实习工作.....因此也特别羡慕那些能拿到大厂offer的同学
- 也是从大三暑期开始，在老师们的指点下慢慢琢磨出了好几个项目，开始用python刷题，搭了个人主页，在Nice实习，寒假参与字节的前端训练营，训练营结束前拿到字节offer.....短短的半个学期，我觉得用“突飞猛进”来形容我自己的进步不算夸张～
- （今年春节假期，一边要完成之前实习留下的几个模型，一边要参与字节前端营，还要准备字节的面试😇，年前搁自习室开启自闭模式，早上九点出门，晚上九点回家，爷快秃

了）

- （我承认我从3月3号一直到3月11号都没好好干活，对不起，我要支棱起来）
- 感觉在北京生活，对能力的要求真的非常高，互联网企业虽然相对高薪，但996文化使得在此发展可能难为长久之计，研究生毕业以后该何去何从，现在还不清晰。目前已经拿到大厂的实习offer，可以提前感受一下整体氛围，为日后打算提供参考；同时考虑研究生期间参与其他企业实习（微软、Shopee等外企重点考虑），或许可以在毕业以前找到合适路线。
- 2021年阶段性目标提前实现，为自己开了一个好头，也希望能保持良好状态，继续提高！

---

## 击不垮我的逆境，终将使我变得更强大

从3月17日入职至今，正好过去了半个月。与当初刚去Nice一样，这一段磨合的时期是接触新知识最频繁的时候，也是进步最快的时候。但与当初不太一样的是，身上的担子重了。从入职第三天开始，mentor就开始给我安排需求了，而在那之前，我接触Go语言不过一周的时间。在字节跳动，用一种几乎没接触的语言，优化一个刚刚上线了的产品，在这之前我只有三天的时间用来配好环境。我坚信入职前我是做了一些困难准备的，只是没想到来得这么快。

第一个需求比较简单，尽管有些惊惧，担心自己修改的部分上线了会出问题，但最后还是比较顺利地交付了。入职一周的时候，我已经提了多次MR。你知道MR是什么意思吗？我当时也不知道，都是现学的（Merge Request）。从拉代码到本地，接优化需求，调整代码，到推上gitlab，发布版本，更新上线，这一系列几乎从没接触过的操作，在mentor给我演示一遍以后，就应该熟悉了。

第二周，有一个难度大一点的需求，需要新建一张表，以及修改一组接口。这段时间我内心仍然是忐忑的，初来乍到，我特别害怕出错，要是我负责上线的版本出了问题不是很难堪。然而越是小心翼翼，越是难以发现问题，战战兢兢提了MR之后马上就会发现其它地方的错误，然后又打回重改，如此往复。

今天是3月31日，这一个需求已经做了一周了，还是没完成。下午在反反复复检查测试以后，同时提了前后端的MR，mentor合的时候可能人都傻了（这学生这么菜是怎么进来的？），我坐在mentor旁边改了一个多小时，还是漏洞百出。当时真的又着急又沮丧，觉得他要求的太严了，我已经把需求实现了，至于出现的问题下次再打个补丁不就好了？改了半天无果，沮丧地回到工位，当时我就想，要不走人事了，别的新同学还有新手任务可以练练手熟悉熟

悉，我倒好，刚来三天就接需求，要求还高的离谱，受这气作甚。

冷静下来，重新审视代码，突然察觉之前写的很多代码是完全无意义无价值的，又仔细考虑了一阵子，做了一个大胆的决定，把所有累赘的代码，统统删了，别的地方要是出了问题再调整。改的过程中我幡然醒悟，自己这段时间太小心了，害怕因为自己的调整把一些看不到的功能改错了，于是说什么都不愿意修改之前的代码，而是新写一个函数，或者结构体来实现更新后的功能，导致Merge的时候各种各样的问题涌现。

意识到问题以后我是欣喜的，沉下心来把冗余的代码一个一个删掉，再重新测试，到晚上功能基本恢复了。

一直觉得我的mentor太push了（我也是最近才体会到什么叫push），本来我就是个新手，这么快就给我提需求，然后还隔三差五来催我这谁顶得住。可是转念一想如果mentor对实习生一点要求没有，前一个月做做新手任务混混时间，真能学到知识吗？我不知道，但我知道绝没有在实战中学的快，学得好。我来这不是来混日子赚生活费的，我是来学技术的。玻璃心，还怎么学技术，只有一个又一个的逆境，才能使我变得更强大。