我对研发22条军规的个人理解以及经历分享

研发22条军规，这是针对研发人员而言，既然是军规，那么制定者就必将希望每个研发人员能始终如一的遵守和执行。这里面每一条军规都非常经典，尤其是以下两条军规，我感触颇深：

> - **万得研发22条军规部分：**
>
> **21.技术攻坚最难的时候正是成长的时候**
>
> **18.让程序运行快一点、再快一点，直到极限**

为什么我对`第21条`和`第18条`军规特别有感触呢？本身我作为一名研发技术人员，应该怀着勇于探索、超越和精益求精的心态去写程序。当你遇到技术难题的时候，首先我们需要停下来思考，是我们的总体技术方案出了问题导致我遇到了技术难题，还是技术方案正确，确实需要技术攻坚。这种思考是必须要，因为技术攻坚可能涉及多人团队，是存在攻坚成本的问题，如果是前者总体技术方案有问题，那么可以考虑重新修正技术方案，那如果是后者，那么这个时候就是需要技术攻坚，并且需要怀着精益求精的精神不断地优化运行速度！

以上这两种情况在我个人的学习过程中都碰到过。我个人喜欢研究爬虫，爬虫在爬取过程中充满了反爬机制，这对于我来说可能是一种反爬的技术攻坚，并且要优化爬虫的抓取速度，我在大学期间也个人也做了很多的爬虫实践。

> **注：以下爬虫经历爬取的数据均属于公开内容，爬取数据仅供学习研究，数据的使用不涉及商业利益**

1. 目前市面上的爬虫框架或者项目基本上基于`Python`语言开发，记得当时在爬取`XX同城`的租房数据时，却选择了`Java`语言进行构建，我们知道现在各大网站都有各种各样的反爬机制，我当时就遇到了一个比较麻烦的问题——页面关键数据进行了**字体加密**，有两种解决方案：

   > 1. 利用`Python`现成的开源`TTFont`字体解析库进行解析破解（背景：已有开源类库，可直接调用）
   > 2. 基于`Java`开发一个`TTFont`字体解析库进行解析破解（背景：开源论坛并没有合适的`Java`类库可以调用，原生的`TTFont`的解析引擎并不能满足需求）

   当时选择了第二种方案：自己尝试开发一个简易的`TTFont`字体破解库，在此之前我对`TTFont`字体解析几乎没有了解，为了能够解决这个加密难点，我尝试了 一个礼拜的资料搜集、代码阅读参考，基于开源的`TTFont`字体解析库实现了一个爬虫反爬`TTFont`字体破解库。可能在大牛眼里，我们某些问题并不难解决，但是如果自己进行了尝试，那么必定会使自己成长。在尝试之前，我不知道自己能不能成功，但是我愿意去尝试，最终成功对`XX同城`的租房信息进行了爬取（当然这个过程里还有其他的反爬机制）。

2. 最近闲来无事在尝试爬取`中国土地市场网`的土地交易数据，前期的反爬分析和反爬机制破解都没有问题，但是这个网站的分页查询仅支持`200页`，我理解这是正常的开发考虑，但是我的爬虫需要将`260万`条数据都抓取下来，那么必须突破查询分页的限制。但是这个网站也考虑到了这点，所以在前端的`JavaScript`代码中进行了一系列的加密，我花了大概一天的时间分析了整个页面的`JavaScript`代码，终于找到了查询分页函数，恰巧翻页的参数仅仅在前端进行了校验，我很顺利的突破了翻页的限制。但代码运行了几个小时以后，我观察到程序抓取的速度越来越慢，从`2条/秒`的速度变成`0.25~0.33条/秒`，这是一个不可接受的抓取速度。经过大量测试，我定位到了问题的根源——网站后台`的SQL查询`耗时较长，导致我的抓取速度变慢。我尝试了集中解决方案，最后

利用时间日期将数据切分成合适的数据块进行爬取，速度也基本稳定在了`1~2条/秒`。这是我觉得最好的解决办法了，如果想更快，那么只能期待网站数据库查询的优化了。

作为技术研发人员，每个岗位都会有技术攻坚任务，如果真的需要技术攻坚，那么就坚持努力地攻破它，并且要不断的对自己的程序性能、速度提出挑战和优化。这是我们自己的成长轨迹，当我们突破了，我们也就成长了。