

Assignment #P: 课程大作业

Updated 2002 GMT+8 Dec 17, 2025

2025 fall, Compiled by 韩旭 元培学院

说明：

关乎每位同学维护自己的 GitHub 作业，本意是让大家练习常用于计算机科学学生的代码和文档维护方法。通过计算机相关课程，我们希望引导大家进入计算机学科领域。这将帮助同学们熟悉实际的编码和文档管理流程，并培养在团队协作和版本控制方面的技能。

- 1) 提交内容，请填写到下面作业模版中。
- 2) 截止时间是期末出分前，因为Canvas可以多次提交，建议期末机考前提交一次，考试后加上课程总结再提交一次。

提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。

评分标准

标准	等级	得分
按时提交	1 得分提交, 0.5 得分请假, 0 得分未提交	1 分
你的GitHub网址	1 得分有, 0 得分无	1 分
你的GitHub截图	1 得分有, 0 得分无	1 分
Cheatsheet	1 得分有, 0 得分无	1 分
课程资料	1 得分有, 0 得分无	1 分
总得分：		5, 满分 5

1. 三个示例

同学开自己的GitHub，自己学习方法、做的题目、考试时候要带的记录纸（cheat_sheet）等放在上面。方便大家关注，当你有新的更新时，我们也可以及时获得最新的内容。

示例：北京大学课程资料整理

<https://github.com/forxhunter/libpku> 这样的项目可以作为一个数算课程的项目，同时也是同学们整理资料的一个好方式，可以实现一举多得的效果。

示例：清华计算机系课程攻略

<https://github.com/PKUanonym/REKCARC-TSC-UHT>

示例：csdiy

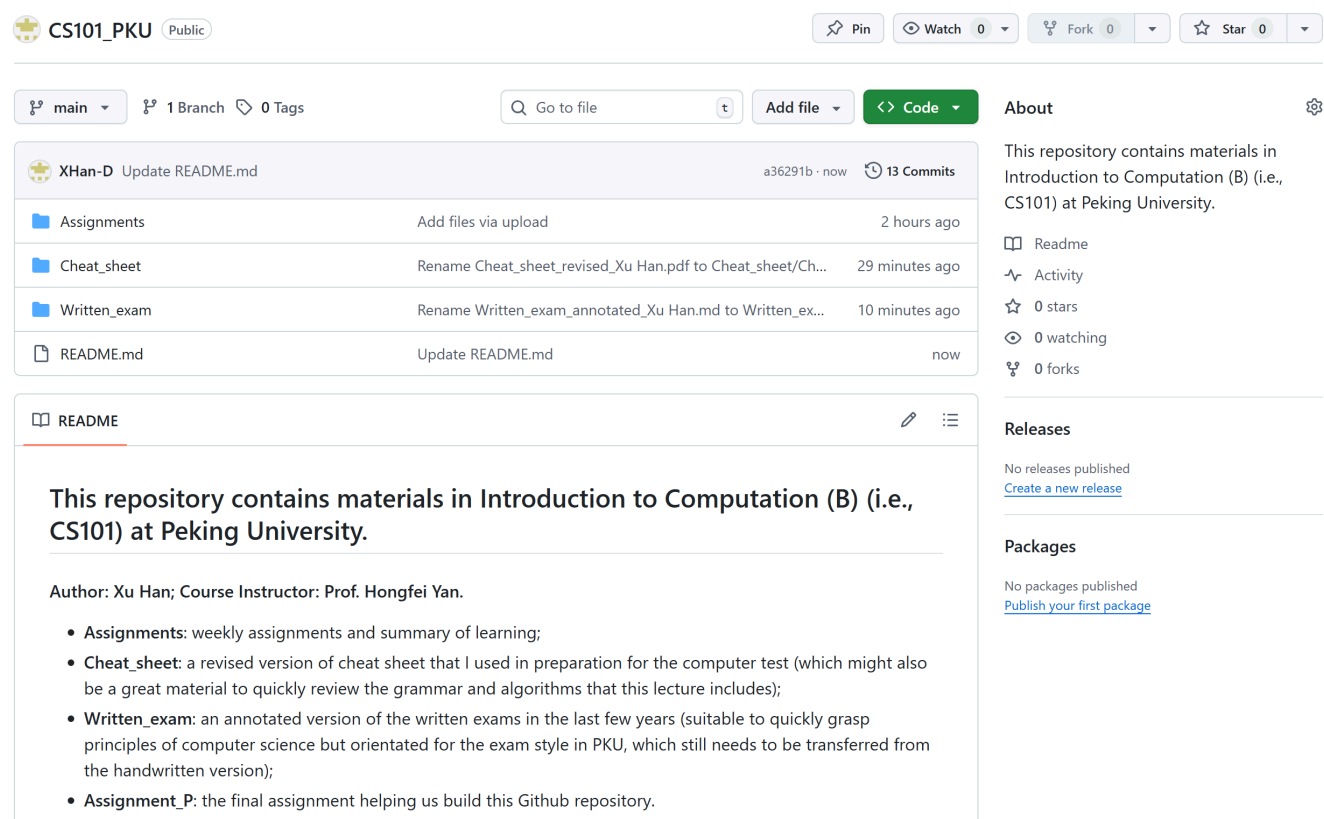
<https://csdiy.wiki/#cs61a>

2. 提交内容

Github网址：

[XHan-D/CS101 PKU: This repository contains materials in Introduction to Computation \(B\) \(i.e., CS101\) at Peking University.](#)

Github截图（Assignment_P还没放进去）：



3. 课程总结

很幸运能在大学的最后一年选到闫老师的计概B。大一的时候由于和专业课撞了没有选计概B，后来大二大三虽然每年都在尝试选但是因为没舍得投点最后也都没选上，幸好大四赶上了闫老师的班扩名额，于是成功选课。我在大一的时候对计算机不仅知之甚少而且有点排斥（虽然我比较喜欢数学）；后来在一些通识课上接触到了一些计算数学和逻辑学的知识（数理逻辑、图灵机等），选了经济学专业之后，又接触到一些计量和统计分析的软件，慢慢觉得编程和算法的知识非常重要（而且也还算有趣）。在高年级进行一些经济学科研的过程中，我对于Stata和R这类软件已经比较熟练了，并且也尝试了用Python进行一些数据分析和模拟。但是这些为心理学和社会科学设计的软件特点就是封装性

比较强，很复杂的算法（比如线性回归）可能一行代码就全部解决了，即使是用pandas这种库做数据分析，所需要掌握的更多是一大堆函数的名字，其实称不上真的会用python。其实我们专业对计概的要求是计概C，但是由于上面这些原因，我觉得应该要上计概B，而且未来的科研需要我学得比较深入，需要我能够在会调用别人写好的函数的基础上自己也会写一些代码甚至是算法。在上这门课之前，我肯定是有一些基础的，但学过以后还是发现不仅语法捉襟见肘（比如对于各种数据结构的相关语法特别不熟练），对于算法更是了解不多（好在这些算法的名字都有所耳闻，学起来没有那么畏难）。

通过一学期的学习，我现在可以比较自信地说对于Python的基本语法是完全没有问题了（至少在允许我查资料的基础上），并且还理解了各种算法的基本思路（虽然对于具体的题目和技巧仍然自我感觉比较生疏）。举个例子来说，我在大学前三年其实听过很多老师和助教提到所谓“动态规划”（因为宏观经济学现在广泛使用DP的思路，这可能是这学期对我的科研最有用的一个算法），但是我其实根本没搞懂过，因为他们要么只是提一下这个名词，要么就是写一些数学公式，我其实不太明白状态、政策、转移方程这些东西都是什么意思。学完计概之后我终于明白DP的大概思路是什么了，其实就是在一个表格里按一定转移规则做一个递推（更多理解也可以看我之前的作业），这的确对我有很大的帮助。另外一个点是，我现在认为编程所谓“熟能生巧”主要是建立不同语法、不同语句、不同结构之间的某种肌肉记忆或者联系，让我在写第一句的时候不需要想半天才能写出来第二句，而是能够把脑子里的数学或逻辑思路快速地转化成编程语言，并且也把编程语言之间快速地建立联系（有点像LLM干的事情）。学期初的时候，我写每段代码的input部分还需要费一些力气思考，现在基本上可以一边写着input部分，一边读题或者思考核心部分的思路了；再比如，写一些相对模板化的算法的时候（比如回溯和BFS），我也已经可以一气呵成地按照题目意思把框架搭建出来。当然，做到这样的程度仍然缺乏灵活性，很多时候我自己的解法相比于更优的解法或者正确的解法之间仍然存在着比较大的差距，比如说没有想到数学上比较简单的递推、没有用到更加方便的record数组、record集合、或者一些数据结构、没有实现有效率的查重和剪枝、或者就是变量命名、代码风格等方面还有待推敲。这些点大概都是在这个学期当中做题或者看题解的时候，我目前还有印象的一些错误和改进。不过由于这学期我需要兼顾课程、科研和毕业论文，所完成的做题量一定没有群里优秀的学弟学妹们那么多的，大概出了基础的作业之外，就是刷了往年考试题、部分每周上机题、每日选做只能挑有空的日子来做，考试前一段时间leetcode又额外做了三十多道。每次月考我大概是AC3，期末机考是AC4，其实第三题比较遗憾，后来复盘是因为查重的地方出了一点bug，但是当时心态比较紧张，没有全面地debug，而是换了一个错误的思路一去不复返了。虽然由于我比较特殊的未来计划，大四的绩点对我而言仍然比较重要，但我想从这门课上学到的编程和算法的知识对于我未来做出更加高效和有原创性的科研工作存在更大的助益的。

虽然大四学习计概的经验对于大一二二的学弟学妹们而言不够适配，但是我想有一些经验教训是共通的。一方面，从我的经历来看，我对计算机和编程的接触有点反过来，本来应该先学计算机基础，然后再进入应用层面，而我其实在应用层面已经用得很熟练了，但是对计算机基础其实知之甚少。首先，我不得不承认，即使一个完全不懂计算机底层的人，在面对被程序员们包装得天衣无缝的统计软件和应用程序的时候也是可以很好地使用的，所以即使计概学得不够好，非计算机专业的同学们仍然可以很好地使用Matlab, R, Stata, LaTeX等等“编程语言”。但是，如果没有语法和算法的基础，这些程序用起来将会非常生硬，我经常遇见我想要实现一些功能，但是已经封装好的程序没有给我提供相关的结构，我去问AI，他建议我自己写一个函数（或者给我一个函数的代码），我要么望而却步，要么用了AI的代码但是它对我来说仍然是黑箱。所以，虽然体贴的程序员们已经为应用计算机的各门科学的学者提供了很大的方便，但如果未来要做出真正原创性的工作，了解底层的逻辑不仅是锦上添花，而且是十分必要的。从另一个方面来看，算法实际上就是解决问题的策略，虽然我们在未来的科研当中恐怕不会遇到算法题这么“简单”（我指的是情境简单）并且经过一定设计的内容，但是这些算法相关的思路往往是屡见不鲜的。仍然以我所在的经济学为例，虽然事实上不了解这些算法并不妨碍我学习经济学（我相信很多其他学科也一样），但是多期决策的宏观经济学所使用的方法本质上就是递归和DP，动态博弈论的均衡解也是递归，社会经济网络当中的很多理论涉及图和搜索，最优化问题的数值解有的时候是可以使用贪心的。这些结论可能是我作为大四学生学完计概之后一些比较独特的总结，但这说明了对于很多专业的学弟学妹而言，或许在学完这门课（和数算）之后，你们大概率不会再如此高强度地在CF、leetcode（和openjudge）上面刷题了，但是我觉得在自己的专业甚至生活当中，尤其是在人工智能和信息化的时代，我们还将很多次地与这些优美的算法相遇。

在更具体的学习经验上面，我自认为掌握得不如很多优秀的同学们那么好，我的很多经验都是从群里面同学们的解法中学习来的，其他同学们的经验也已经比较完备了。但我确实有两点感悟比较深的地方：第一，编程相对而言可能比数学、物理和文字性的东西更加陌生，因此也更会让人有畏难情绪，但是我觉得闫老师班的节奏特别好地让我们理解了，编程语言本身并没有什么难度，随着接触的语言和软件越来越多，其实这些语言的核心部件大部分都是一样的（就拿R来说，虽然没有python那么灵活，但是其基本语法也都是数据类型、初始化、顺序分支循环结构、函数这些；Matlab也一样；Stata就更加简单了，只有基本的句法结构和循环语句）。我觉得用学习Python语法的这种效率和精神来学习很多新的东西（包括闫老师要求我们学习的Shell，Github，Markdown，以及AI的使用，这些也是一样），并且逐渐克服对接触陌生事物的畏难情绪，是闫老师班上的同学们很重要的一点收获；第二，学习题解或者其他人的解法是非常重要的，但是要找到自己的节奏。比如说自己先AC了还是否需要看题解呢？自己WA了很多次要不要直接看题解呢？如果题解的思路和我之前的思路完全不同，我要不要继续自己原来的思路呢？我觉得这些都需要在开始学算法的那几周慢慢理解。我的经验是，前期即使自己AC了也一定要看题解，因为可能包括很多自己不会但可以用比较笨的办法搞定的小技巧（比如map, join这类），后期如果自己AC了也可以不看题解；如果WA了很多次，可以考虑先问AI，AI会帮助你按照你已有的思路指出问题，但一定要和AI充分讨论，不能言听计从，相当于找了一个同学来帮你一起解决这个问题。对于这种题，无论如何最后一定要看一下题解，有全新的方法就记录下来，然后有时间的话可以自己写一遍（我印象比较深的是购物、排队、红蓝玫瑰这几个题，我是看懂了然后自己写了好几遍的）。如果你想要在最终考试中表现得比我更好，那么可以突破的点可能就是多刷题来提升速度和熟练度，并且掌握更丰富的debug技巧和考试节奏的技巧。最后就是如果你和我一样在这门课程中有一些遗憾的地方，也没有关系，一方面我们应该沿着这门课指出来的路径，各取所需地进一步深化我们的相关技能；另一方面，或许某一道题没有AC、某一门课没拿到满意的成绩在局部看来的确是错失了最优解，但是人生的结构多么复杂呀，用贪心来解肯定会出错的。

参考

1.科学上网 Scientific Internet

北大学长提供的Clash，请自己取用。

<https://189854.xyz/verify/>

<https://blog.189854.xyz/blog/walless/2023/11/04/clash.html>

2.图床，把图片放到云上去，而不是本地的意思。如果设置图床，分享md文件，其他人也能看到图片；否则因为md嵌入的图片在本地，只有编辑者能看到；后者的情况解决方法还可以是导出包含图片的pdf文件分享。图床如果是免费的，过一阵可能会失效，之前用过非github的免费图床，导致链接失效了。github是免费的，目前比较稳定。

1) Typora + GitHub = 效率，<https://mp.weixin.qq.com/s/hmkGZIn-xatrWrBZrY9t-g>

2) Typora+PicGo+Github解决个人博客图片上传问题 <https://zhuanlan.zhihu.com/p/367529569>

3) 设置的图床目录是Public

3.Github图片不显示，原因是DNS污染。两种解决方法，或者直接添加1) 给出的ip列表，或者2) 自己找出ip添加。

1) Github图片显示不出来？两步解决！https://zhuanlan.zhihu.com/p/345258967?utm_id=0&wd=&eqid=ce16938700061ac4000000056470d782。

2) <https://www.ipaddress.com>查到ip，添加到hosts后，在移动宽带网络中，可以显示md中的图片。参考：解决raw.githubusercontent.com无法访问的问题（picgo+github配置图床图片不显示，但仓库已存储成功），<https://blog.51cto.com/reliableyang/6457392>。