

Assignment #P: 课程大作业

Updated 1141 GMT+8 May 27, 2025

2025 spring, Compiled by ==汤伟杰，信息管理系==

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。通过计算概论、数据结构和算法等课程，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) <mark>截止时间是期末出分前 4 天</mark>，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。

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 and 情报	1 得分有，0 得分无	1 分
总得分：		5，满分 5

1. 要求

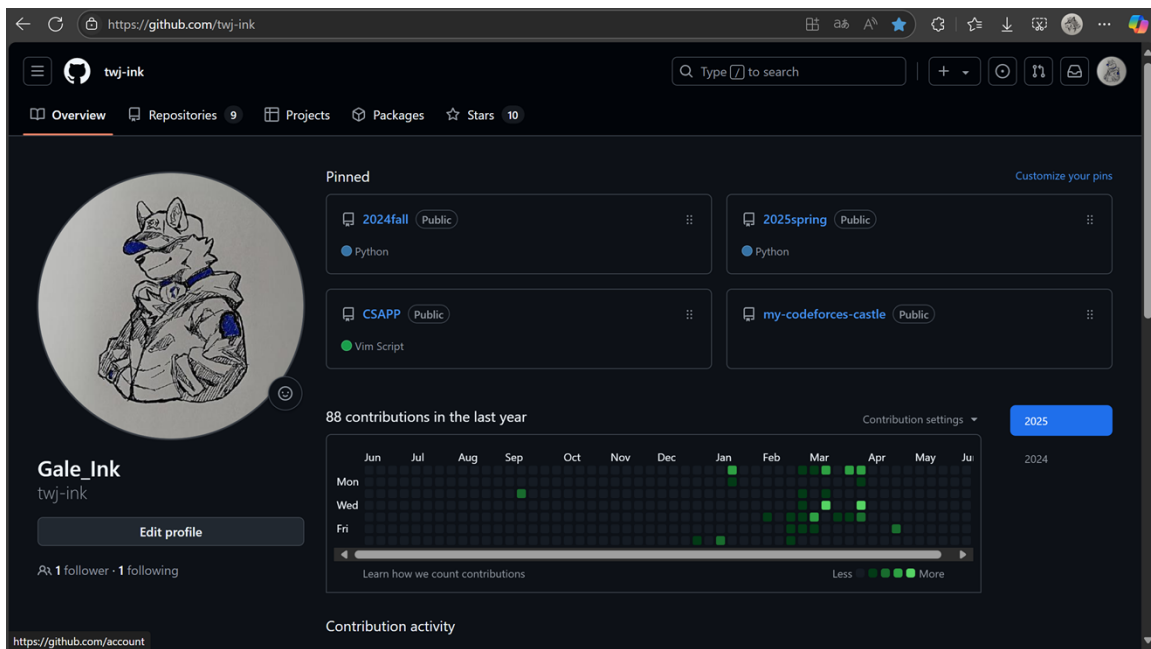
同学开自己的 GitHub，自己数算的学习方法、做的题目、考试时候要带的记录纸（`cheat_sheet`）等放在上面。方便大家关注，当你有新的更新时，我们也可以及时获得最新的内容。

2. 提交内容

<mark>你的 GitHub 网址及截图</mark>。

网址：[汤伟杰的 github](https://github.com/twj-ink)
<https://github.com/twj-ink>

截图：



3. 课程总结

如果愿意，请同学或多或少做一个本门课程的学习总结。便于之后师弟师妹跟进学习，也便于教师和助教改进教学。例如：分享自己的学习心得、笔记。

又到了写期末总结的时候了，这个学期数算的学习和计概的学习体验其实是有诸多不同的。

先说说数算课程的学习体验吧，在寒假的时候整天呆在家里，大部分时间其实都在啃老师提前发出来的课件，并且最终啃完了线性结构、树这两个课件，图的课件在寒假时没有细细啃，只是粗略看了一眼，发现 `dfs` 啊 `bfs` 啊 `dijkstra` 啊好像都是计概里面有涉及的部分。所以在正式开学之后的前几个月，我实际上在数算课件学习上没有投入时间，因为寒假的时候已经研究过了，比如树的建树过程，各种遍历方式，链表的一些固定写法等等。然后在课程后期，我才开始具体啃图的课件，并且在几天时间内把课件内容进行了总结汇总，发现模板性的的确确是很强的，代码写起来都大差不差。在期末时候老师出了一些树形 `dp` 的问题，我便开始学习理解这种 `dp` 的思维，即往往是从树叶向根的填充方式，而这又是借助 `dfs` 的回溯过程完成的。这其实又回到了计概的思维了。最终期末考试 `ac` 了 5 道题，比较吃亏的是第三题，一开始采用逐个字符遍历的方式，一直 `wa`，在最后 30min 换了一种策略，采用 `split('/')`，这才 `ac`，于是第五题 `debug` 的时间就比较少了，再加上通过率很低，导致心态不稳，最终还是以 5 道题收尾。比较难受的是没办法双百了，我的双彩虹呜呜呜。。

然后再说说这个学期的算法体验，由于刚开学时课程内容已经学了一部分，于是我这学期花了很多的时间在 `codeforces` 上打比赛，目前已经 1350 分了，下次再打一场 `div3` 估计有希望从绿名变成青名。在计概时候打比赛其实比较痛苦，因为往往只能做出来 A 题或者甚至一道题都做不出来；而这学期可能是因为心态比较好吧，发挥的比上学期要更好一点，有时甚至能在 `div2` 中做出 3 道题，很开心很开心。而 `cf` 上的题目特点其实与力扣差别很大，`cf` 上题目的思维性往往比较强，力扣上比较专注于数据结构和算法。最鲜明的一类 `cf` 上的题目是打上了 `constructive algorithm` 构造算法标签的题目，这种题目的思维性一般都很强，做起来也十分有趣，不过难度也往往很大。

结果就是这学期在力扣上刷题比较少，而且刷题的目的往往是针对某个知识点去具体刷，比如后期学习 **kmp** 算法、**bellman** 算法等等时候去具体刷题。期末考试遇到了力扣风格的题目，说实话内心还是有点畏惧的，因为风格原因害怕考场上做不出来，不过最终老师选的题目还算比较温和，也是做出来了。

此外，我整理了这学期后半程 **cf** 几乎所有比赛的一些比较好的题目（因为后半程的每一场我几乎都有参加），放在这里供老师选题用（感觉有些题目真的很好，很适合给下一届计概学生做，或者考试做，或者数算学生做）。我也会继续整理一些比较好的 **cf** 比赛题放到 **github** 中，采用和老师类似的结构，也算是 **cf** 选做了。

链接：[cf 好题列表](#)

其中 [E. Kirei Attacks the Estate](#) 是一个典型的树形 **dp**；

[D. Fewer Batteries](#) 是一个二分+图的题目，这两个题目都很适合做数算的 **tough** 级别的题目。

此外，说说这学期接触到的一些其他关于编程的东西。我选了程设这门课，前半学期突击学习了 **cpp** 语言，说实话刚开始学的时候我的内心是极其痛苦的，既没有 **py** 简洁，又没有 **py** 简单，各种关键词满天飞，大括号缩进看的挺难受的。不过我看网上吐槽真正做起项目的时候 **py** 的“游标卡尺式”缩进可能不是很好，但是从我写代码来看，**cpp** 除了在时间上有着巨大的优势之外，其简洁性和易写性都不太比得过 **py**，当然这也有可能是因为我目前还不是非常熟练导致的。不过用 **cpp** 总是存在内存爆炸的情况，特别是做树的题目，需要用一堆指针，如果一个题目的一次测试有多组数据，要建多次树，如果不把上次的所有节点都 **delete** 掉，最后极有可能喜提 **MLE** 而 **debug** 半天。在期末的时候无意接触到 **lean4**，这是一个数学证明形式化的语言，目前还处于招募人做数据标注的阶段，我也极速突击了一下，感觉这种语言和其他编程语言的区别还是蛮大的，目前也是不太熟练，处于极其痛苦的阶段。不过和写算法题一样，最后命题得到证明之后的喜悦和 **ac** 的喜悦是一样的。

最后，感谢计概的 **cs101** 和数算的 **cs201**，感谢闫老师的算法启蒙，让我有勇气去参加转专业的机考，并最终获得了转入信科的资格。希望自己还能带着大一的这份钻研算法的精神继续努力下去！

4.cheatsheet 和课程情报

在期末极速整理了一个全是板子的带到考场的 **cheatsheet**：

[final cheat sheet](#)

但日常的一些比较好的题目、按类型整理的一些题目汇总在这里：

[2025spring good questions](#)

以及关于图的汇总：

[graph](#)

和 **kmp** 算法的汇总：

[kmp](#)

关于课程情报，课件内容存放在这里：

[课程内容情报](#)