

Assignment #P: 课程大作业

Updated 1818 GMT+8 Nov 30, 2024

2024 fall, Compiled by 李承容 物理学院 2400011499

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 and 情报	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。

github.com/forxhunter/libpku

libpkuPublic

forked from lib-pku/libpku

Watch0

Fork0

Star0

master1 Branch0 Tags

Go to file

Add file

Code

This branch is 93 commits behind lib-pku/libpku:master.

librarian-pkuUpdate README.md6185ca6 · 5 years ago39 Commits

专业课	Add files via upload	5 years ago
公选课/Python编程艺术	Add files via upload	6 years ago
政治课	Add files via upload	6 years ago
暑期课/acm-icpc暑期课	Add files via upload	6 years ago
英语课	Add files via upload	6 years ago
通选课	Add files via upload	6 years ago
.gitignore	Add files via upload	6 years ago
.travis.yml	Update .travis.yml	6 years ago
README.md	Update README.md	5 years ago
index.js	Update index.js	6 years ago

README

libpku - 北京大学课程资料整理

Preface

(引用自 QSCTech/zju-icicles)

来到一所大学，从第一次接触许多课，直到一门一门完成，这个过程中我们时常收集起许多资料 and 情报。

有些是需要在网上搜索的电子书，每次见到一门新课程，Google 一下教材名称，有的可以立即找到，有的却是要花费许多眼力；有些是历年试卷或者 A4 纸，前人精心收集制作，抱着能对他人有用的想法公开，却需要在各个群或者私下摸索以至于从学长手中代代相传；有些是上完一门课才恍然大悟的技巧，原来这门课重点如此，当初本可以更轻松地完成得更好.....

About

北京大学课程资料整理

[lib-pku.github.io/](#)

Readme

Activity

0 stars

0 watching

0 forks

Report repository

Releases

No releases published

Packages

No packages published

Languages

JavaScript 100.0%

示例：清华计算机系课程攻略

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

github.com/PKUanonym/REKCARC-TSC-UHT

大一上	2024.2.7更新 (#190)	last week
大一下	2024.2.7更新 (#190)	last week
大一小学期	2024.2.7更新 (#190)	last week
大三上	2024.2.7更新 (#190)	last week
大三下	数学实验 (#175)	8 months ago
大三小学期	readthedocs 1st try	4 years ago
大二上	2024.2.7更新 (#190)	last week
大二下	2024.2.7更新 (#190)	last week
大二小学期	Fix readme (#169)	8 months ago
大四上	数字图像处理 & 媒体计算 (#172)	8 months ago
研究生/组合数学/exam	cmt	5 years ago
.gitignore	readthedocs 1st try	4 years ago
LICENSE	LICENSE (#24)	5 years ago
Makefile	readthedocs 1st try	4 years ago
README.md	2024.2.7更新 (#190)	last week
make.bat	readthedocs 1st try	4 years ago
setup.py	readthedocs 1st try	4 years ago
参考书目.md	更改文档结构 & 微积分 & 线性代数 & 汇编 & 自动机 & 复变 ...	3 years ago
收录内容.md	大一小学期 & 软件工程 & 虚拟现实技术 (#129)	2 years ago
贡献方法.md	readthedocs 1st try	4 years ago

README

CC-BY-SA-4.0 license

清华大学计算机系课程攻略

示例：csdiy

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

CS自学指南

前言

如何使用这本书

一个仅供参考的CS学习规划

必学工具

好书推荐

数学基础

数学进阶

编程入门

电子基础

数据结构与算法

软件工程

计算机系统基础

体系结构

操作系统

并行与分布式系统

计算机系统安全

计算机网络

数据库系统

编译原理

编程语言设计与分析

计算机图形学

Web开发

数据科学

人工智能

机器学习

机器学习系统

深度学习

机器学习进阶

后记

梦开始的地方 —— CS61A

大一入学时我是一个对计算机一无所知的小白，装了几十个 G 的 Visual Studio 天天和 OJ 你死我活。凭着高中的数学底子我数学课学得还不错，但在专业课上对竞赛大佬只有仰望。提到编程我只会打开那笨重的 IDE，新建一个我也不知道具体是干啥的命令行项目，然后就是 `cin, cout, for` 循环，然后 CE, RE, WA 循环。当时的我就处在一种拼命想学好但不知道怎么学，课上认真听讲但题还不会做，课后做作业完全是用时间和它硬耗的痛苦状态。我至今电脑里还存着自己大一上学期计算概论大作业的源代码 —— 一个 1200 行的 C++ 文件，没有头文件、没有类、没有封装、没有 unit test、没有 Makefile、没有 Git，唯一的优点是它确实能跑，缺点是“能跑”的补集。我一度怀疑我是不是不适合学计算机，因为童年对于极客的所有想象，已经被我第一个学期的体验彻底粉碎了。

这一切的转机发生在我大一的寒假，我心血来潮想学习 Python。无意间看到知乎有人推荐了 CS61A 这门课，说是 UC Berkeley 的大一入门课程，讲的就是 Python。我永远不会忘记那一天，打开 CS61A 课程网站的那个瞬间，就像哥伦布发现了新大陆一样，我开启了新世界的大门。

我一口气 3 个星期上完了这门课，它让我第一次感觉到原来 CS 可以学得如此充实而有趣，原来这世上竟有如此精华的课程。

为避免有崇洋媚外之嫌，我单纯从一个学生的视角来讲讲自学 CS61A 的体验：

- 独立搭建的课程网站：一个网站将所有课程资源整合一体，条理分明的课程 schedule、所有 slides, homework, discussion 的文件链接、详细明确的课程给分说明、历年的考试题目与答案。这样一个网站抛开美观程度不谈，既方便学生，也让资源公正透明。
- 课程教授亲自编写的教材：CS61A 这门课的开课老师将 MIT 的经典教材 *Structure and Interpretation of Computer Programs* (SICP) 用 Python 这门语言进行改编（原教材基于 Scheme 语言），保证了课堂内容与教材内容的一致性，同时补充了更多细节，可以说诚意满满。而且 [全书开源](#)，可以直接线上阅读。
- 丰富到让人眼花缭乱的课程作业：14 个 lab 巩固随堂知识点，10 个 homework，还有 4 个代码量均上千行的 project。与大家熟悉的 OJ 和 Word 文档式的作业不同，所有作业均有完善的代码框架，保姆级的作业说明。每个 Project 都有详尽的 handout 文档、全自动的评分脚本。CS61A 甚至专门开发了一个 [自动化的作业提交评分系统](#)（据说还发了论文）。当然，有人会说“一个 project 几千行代码大部分都是助教帮你写好的，你还能学到啥？”。此言差矣，作为一个刚刚接触计算机，连安装 Python 都磕磕绊绊的小白来说，这样完善的代码框架既可以让你专注于巩固课堂上学习到的核心知识点，又能有“我才学了一个月就能做一个小游戏了！”的成就感，还能有机会阅读学习别人高质量的代码，从而为自己所用。我觉得在低年级，这种代码框架可以说百利而无一害。就是苦了老师和助教，因为开发这样的作业可想而知

目录

[梦开始的地方 —— CS61A](#)

为什么写这本书

自学的好处

自学的坏处

这本书适合谁

特别鸣谢

你也想加入到贡献者的行列

关于交流群的建立

请作者喝杯下午茶

2. 提交内容

<https://github.com/licr-fwd/Assignment-P-by-lcr/upload/main>

github.com/licr-fwd/Assignment-P-by-lcr

检测到当前不是目标语言网页，是否需要翻译为：简体中文 [翻译网页](#) 剩余次数：5次 请登录后继续使用 ~ [登录](#) ☐ 自动翻译当前网站

Assignment-P-by-lcr Public

main 1 Branch 0 Tags

Go to file Add file <> Code About

licr-fwd	Update README.md	b0779f7 · now	5 Commits
Cheat Sheet go!.pdf	Add files via upload	7 minutes ago	
Cheat Sheet.md	Add files via upload	7 minutes ago	
README.md	Update README.md	now	
python习题经验总结.md	Add files via upload	7 minutes ago	
python习题经验总结.pdf	Add files via upload	7 minutes ago	

README

Assignment-P-by-lcr

ChengRong Li's assignment P

The following files consist:

- *cheatsheet of the final exam(named as cheat sheet)
- *summaries of part of the practises done
- *file of assignment P(md and pdf)

sincerest gratitude to teacher and TAs

ChengRong Li's assignment P

- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

3. 课程总结

1.一切仿佛都离我远去了，我还在向前奔跑，回忆却停留在原地。

2.期末幸运的AK了，权衡各方面效益决定战术性放弃笔试吃机考的大保底，可惜最后还是落得一无所获的下场，不过已是习以为常了，能有机考这一次的幸运仿佛已经足够了。关于一路走来的一切，有千言万语想要写下，遂另作文章，不再此处赘述了。

3.收获的一切仿佛已经不能用短短的几行文字所描述了吧。课程情报的话，笨人习惯于用老师的课件自学，经常连回放都忘记看，还是非常感激老师灵活的授课方式，没有要求大家必须每节课都坐在教室里。课件的内容层次分明，内容丰富，虽然有的时候没有详细的知识点讲解，但通过研究例题里面的代码注释其实已经能够把算法的思路理解的七七八八了，再通过例题进行巩固，绝大多数问题基本也就研究清楚了。

算法和数据结构方面，最基本的递归，dp，贪心，搜索不必多说，数据结构如果认真了解的话也有很多，栈，堆，树。其他一些高级的算法技巧比如并查集，滑动窗口，dilworth定理，dijkstra，kadane，huffman也都能在讲义中学到，但更多的其实是理解以及触类旁通。很多问题其实是从原理上有一定相似之处的，比如拦截导弹，kosuke's assignment，完美的爱，愉悦旋律，前面两个从tag上就是dilworth定理，而后面两个则运用了类似的思想。再比如树的思想，剪绳子是标准的huffman tree，但实际上波兰表达式的倒序栈写法何尝不也是一种树的想法（虽然本蒟蒻对树的理解还不是很好）。

4.说白了，算法其实无非是一种思维方式。生活中，我们往往对很多问题有一些直观逻辑，但这些底层逻辑有的时候很难被触碰到。比如考虑一个简单的问题：如果高考想考高分，该怎么办呢？我们习以为常的会想到，只要每门都考满分就好了，而这又何尝不是一种贪心算法（名副其实的贪心，大佬除外）——把高考划分为六个子问题，每个子问题都达到局部最优（满分），那么高考自然是最高分。再比如，期末复习的时候我们有很多门课需要复习，我们自然会有两种复习方式，几门课同时复习，或是一门复习完再去复习另一门，我想没有什么比这更显而易见了（确实，废话文学），但这不正是深度优先搜索以及广度优先搜索吗。这些算法的引入并不是空穴来风，甚至算不上是模板，他是我们正常的思维方式，那些数据结构以及基本逻辑结构的选择无非是帮助我们将脑子中的逻辑结合实际情形实现成计算机可以理解（编译）的语言，而选用不同的语言就像是中文，英语，日语，法语一样，他们传达的意图是一样的，只不过语法和词汇有所差异。再说算法的选择，以上面复习的例子为例，中学阶段由于考试基本都是集中在几天考完，因此广度优先复习法显然是更受推崇的，而到了大学，由于两场考试间隔时间很长，因此深度优先复习法显然优势更明显。结合不同的因人而异的情境（最短非挂科复习途径，总绩点的最大值复习途径），以及个人能力（不超时，不超脑容量），进行适当优化（利用cheatsheet缓存（记录不可变类型——模板或是知识点），用todolist作为双向队列按时间顺序记下复习安排然后边做边删除（有滑动窗口那味了）），欸嘿，这就复习完了。当然，每个人的cpu差异还是有明显差别的（比如我就是2核小废物），有的人不需要怎么优化就可以复习完并考得很好，而有些人可能进行了无数剪枝依旧杯水车薪（比如我就剪掉了计概笔试，bushi我不是故意的，但显然，这种常数级的优化并不能带来明显的效果）。

5.最后的最后，感谢老师和助教们的悉心付出与殷殷教诲，言犹在耳，情岂忘心。还不知道未来的路如何，但一路走来的回忆是切实的，那个最能在每次作业后面废话的人，在这最后一次作业里竟表现出了前所未有的克制——说是克制，到头来也还是废话连篇，无痛呻吟罢了。这世界上或许最是敏感的人最是难得归宿，留恋美好，铭记伤痛，憧憬未来，彷徨前途，跌跌撞撞哭哭啼啼地走遍人间的风雪依稀，兰叶葳蕤，领略暮色垂浮，夕云乱卷，噫，古来骚客，莫不得志一而失意殊，凭临天地，却道是也不过漫嗟荣辱罢。

后之览者，亦将有感于斯人。

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/>

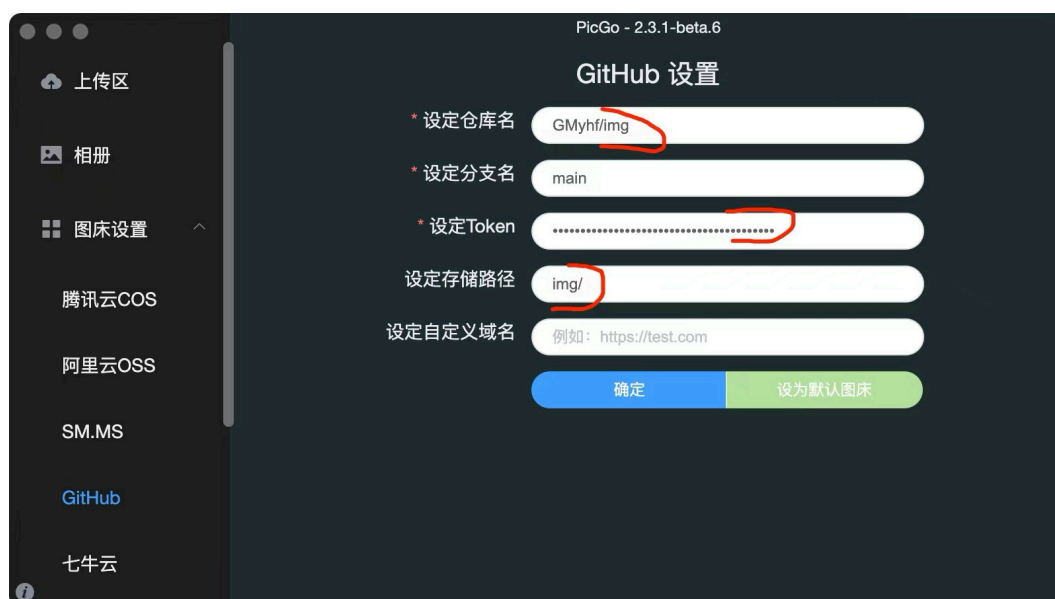
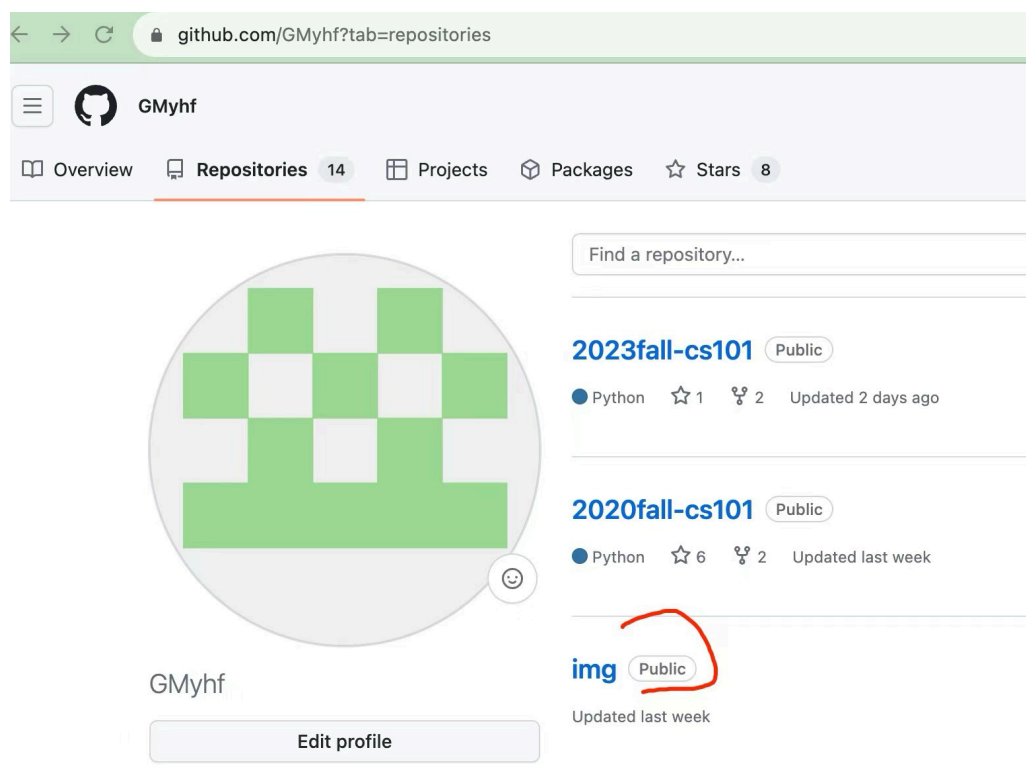
<https://blog.189854.xyz/blog/walless/2023/11/04/clash.html>

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/hmkGZln-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>。