

# 20240910-Week1-计算概论课程概述

Updated 1403 GMT+8 Sep 10, 2024

2024 fall, Complied by Hongfei Yan

本周发布作业：

assignment1, assignmentP

<https://github.com/GMyhf/2024fall-cs101/tree/main/homework>

作业评分标准

标准	等级	得分
按时提交	提交得1得分, 请假0.5分, 未提交0分	1 分
源码、耗时（可选）、解题思路（可选）	4或4+题目得1得分, 2或2+题目0.5分, 无源码0分	1 分
AC代码截图	4或4+题目得1分, 2或2+题目0.5分, 无截图0分	1 分
Canvas清晰头像、pdf、md/doc	三项全得1分, 有二项0.5分, 少于二项0分	1 分
学习总结和收获	有得1分, 无0分	1 分
总得分： 5 , 满分 5		

# 一、课程安排和内容

学习计算机科学与掌握其他高难度学科没什么不同。成功的唯一途径便是循序渐进地学习其中的核心思想。刚开始接触计算机科学的人，需要多多练习来加深理解，从而为学习更复杂的内容做好准备。此外，初学者需要建立起自信心。

“计算概论”，即“算法与数据结构”，通常是计算机科学专业的第一门课程，相较于第二门课程“数据结构与算法”，更强调算法。同学们已经准备好进一步探索这一领域并且进一步练习如何解决问题。

本课程旨在培养计算思维，包括数学思维和计算机思维。这种思维的培养需要结合实践过程中的思考和感悟。同学们将通过类似于阅读说明书的方法快速掌握编程语法，然后在应用算法解决实际问题的过程中深入体验计算思维的魅力。为此，我们将练习100余道习题，并提供简洁的解题思路供参考和比较。

课程知识点涵盖两部分。首先是培养学生对计算机的兴趣和学习具有重要意义的三个计算机原理。其次，重点是打牢编程基础，掌握各种算法的应用实践。具体来说：

第一部分，从计算机文化历史开始，讲到三个主要的计算机原理：图灵机、进程的虚拟地址空间和 ASCII 表。旨在激发同学的学习兴趣，对计算机建立感性和理性认识。

第二部分包括基本语法学习和编程练习。首先，学习 Python 和 C++ 语法，掌握与计算机进行交流的语言，侧重于讲授 Python 实现的代码，在两个编程平台上 ([codeforces.com](http://codeforces.com), [cs101.openjudge.cn](http://cs101.openjudge.cn)) 运用多种常见算法解决问题。一旦掌握了语法，会进一步练习更多的编程题目。

我们的做题平台主要使用 OJ (<http://cs101.openjudge.cn>) 和 CF (<https://codeforces.com/>)。也欢迎使用 Luogu (<https://www.luogu.com.cn>) 和 LeetCode (<https://leetcode.cn>)。特别是 OJ，对于后续的《数据结构与算法B》课程来说非常重要，几乎所有的班级都在使用它。

Python是一门非常适合于讲解算法的语言，语法干净简洁，用户环境直观，基本的数据类型十分强大和易用。

我们相信，对于初学者来说，投入时间学习与算法和数据结构相关的基本思想是非常有益的。

TIOBE 2024 年 09 月份的编程语言排行榜，<https://www.tiobe.com/tiobe-index/>

Sep 2024	Sep 2023	Change	Programming Language	Ratings
1	1		Python	20.17%
2	3	▲	C++	10.75%
3	4	▲	Java	9.45%
4	2	▼	C	8.89%

## Programming Language

- The language should have an own entry on [Wikipedia](#) and Wikipedia should clearly state that it concerns a programming language.
- The programming language should be [Turing complete](#).

- In [computability theory](#), a system of data-manipulation rules (such as a [model of computation](#), a computer's [instruction set](#), a [programming language](#), or a [cellular automaton](#)) is said to be **Turing-complete** or **computationally universal** if it can be used to simulate any [Turing machine](#)[\[citation needed\]](#) (devised by English mathematician and computer scientist [Alan Turing](#)). This means that this system is able to recognize or decode other data-manipulation rule sets. Turing completeness is used as a way to express the power of such a data-manipulation rule set. Virtually all programming languages today are Turing-complete. 翻译：在[可计算性理论](#)中，一个数据操作规则系统（如[计算模型](#)、[计算机的指令集](#)、[编程语言](#)或[细胞自动机](#)）如果能够用来模拟任何[图灵机](#)（由英国数学家兼计算机科学家[艾伦·图灵](#)发明）则被称为图灵完备或计算通用。这意味着该系统能够识别或解码其他的数据操作规则集。图灵完备性被用来表示这种数据操作规则集的能力。几乎当今所有的编程语言都是图灵完备的。
- 推荐一个电影，The Imitation Game，是图灵自传。<https://pku.instructuremedia.com/embed/6f8ab27f-9485-497e-a8ae-24f8be176f22>
- 布莱切利园（Bletchley Park）之旅报告，郭媚妤，2022年11月23日，[https://github.com/GMyhf/2024fall-cs101/blob/main/Report\\_on\\_the\\_Visit\\_to\\_Bletchley\\_Park.pdf](https://github.com/GMyhf/2024fall-cs101/blob/main/Report_on_the_Visit_to_Bletchley_Park.pdf)
- The programming language should have at least 5,000 hits for +"programming" for Google.

## 1.1 课程安排

### 1.1.1 线下和线上教室

课程号和名: 04831410, 计算概论(B)	班号: 11
上课时间: 1-16周 每周 周二 7-9节	线下地点: 二教107
上机时间: 2-15周 每周 周四 7-8节 期末机考时间: 2024.12.26 周四 7-8节 期末笔试时间: 2025年1月7日下午	地点: 7号机房 (理科1号楼三层1339房间) 和8号机房 (1338房间)
助教: 熊江凯、罗熙佑、涂程颖、王嘉林、刘昊文	助教在微信群中的名字是“TA-”开始, 地点: 理科1号楼1220

主题: 2024fall-cs101-ADS 计算概论 (B)

时间: 2024年9月~12月, 每周二15:00~18:00 北京

加入Teams会议:

[https://teams.microsoft.com/l/meetup-join/19%3ameeting\\_MGVhMWY3NmUtMGUxNi00YTM4LThkNDAtOGRhMjRkMjFiNmJl%40thread.v2/0?context=%7b%22Tid%22%3a%22cd66712e-8aa9-44ad-9e42-02ea2d037e64%22%2c%22Oid%22%3a%227fae1325-5f9b-401d-bd3f-42dd9a7d27b8%22%7d](https://teams.microsoft.com/l/meetup-join/19%3ameeting_MGVhMWY3NmUtMGUxNi00YTM4LThkNDAtOGRhMjRkMjFiNmJl%40thread.v2/0?context=%7b%22Tid%22%3a%22cd66712e-8aa9-44ad-9e42-02ea2d037e64%22%2c%22Oid%22%3a%227fae1325-5f9b-401d-bd3f-42dd9a7d27b8%22%7d)

Meeting ID: 467 329 315 882

Passcode: zSs3ve

没有签到, 可以线下、线上、回看方式学习。

## 1.1.2 2024 fall 上机说明

时间：2-15周每周四7-8节（15:10~17:00）。

地点：7号机房（理科1号楼三层1339房间）和8号机房（1338房间）。

使用机房电脑：打开电脑后，用学号和网关密码登陆，机房电脑有pycharm、anaconda/spyder和vscode。

同学可以选择去机房完成作业或其他编程练习，暂时没有固定座位，机房会有助教为大家答疑。

有助教因为校区较远，可能无法每次都线下参与，其余时间会以线上会议的方式参与上机课的答疑。如果学生不多，答疑有可能会集中在一个机房进行。

建议大家充分利用上机时间集中答疑，避免其他时间1对1找老师或助教debug。

另外：

- 1) 网络问题，计算中心 62751023；
- 2) 机房上机，账号/登录等问题，计算中心值班老师。

机房二楼值班室：理科一号楼1254W，电话：62753392

机房三楼值班室：理科一号楼1343W，电话：62753352

## 1.1.3 期末考试时间

平时上机不来不影响成绩，但是期末有机考+笔试，都必须参加。

期末机考时间：第16周上机时间，即2024.12.26 周四 7-8节。往年是6个题目，需要在机房完成。

期末笔试时间：2025年1月7日下午。往年平行班笔试包括：填空15分、单选15分、计算题20分，编码应用题12分，程序阅读12分，程序填空10分，流程图与编程12分，问答4分。

我们班往年是提高班，今年2024fall是平行班，所以增加了笔试。

万一考试时间冲突了怎么办？只能同学自己协调，极端情况是：去年有化院同学只能参加30分钟考试。没有补考。

## 1.1.4 答疑安排

线上7\*24小时答疑：课程微信群。

通常提问给出：题目链接，清楚截图，遇到问题的描述，编写的源码（便于TA拷贝来本地测试）。

几个基础的报错：compile error是编译根本不成功，presentation error大家可能要考虑是不是结果对了但输出的形式有问题，比如多一个空格少一个空格，runtime error运行时错误一般是考虑除0，数组越界，输入为空这种情况，time limit exceeded和memory limit exceeded分别是爆时间和爆内存这时候要去考虑算法优化，wrong answer是结果错误，这种情况除了考虑正常样例之外可能还要检查设计逻辑，构造corner case，除了尝试题目里的样例还可以自己丢进去试一下。

线下答疑：

上机时间：助教轮流现场答疑。2-15周

助教办公室答疑时间，地点在理科1#1220E，2-15周

罗熙佑：周一，19:00-20:00

涂程颖：周二，19:00-20:00（线上）

刘昊文：周三，19:00-20:00

熊江凯：周四，19:00-20:00

王嘉林：周五，19:00-20:00

## 1.2 课程内容

没有固定教材，会全面介绍在经典计算机科学问题中出现的算法和必要的数据结构。尽管在学习顺序上并无严格要求，但是许多话题之间都存在一定的依赖关系，建议顺序学习。

课程内容推进安排：首先定位是平行班，9月份以同学自主学习编程语法，10月份贪心、动态规划，11月份递归、搜索，12月份复习，1月份优秀！下表是具体16周的课程内容：

1. 课程概述	2. 计算机文化	3. 开发环境和程序设计语言
4. 计算机原理	5. 语法（1/2）：变量、操作符	6. 语法（2/2）：控制结构
7. 算法（1/6）：贪心和动态规划	8. 算法分析	9. 信息传递和二维数组
10. 算法（2/6）：递归和动态规划	11. 算法（3/6）：递归和动态规划	12. 算法（4/6）：搜索（1/3）
13. 算法（5/6）：搜索（2/3）	14. 算法（6/6）：搜索（3/3）	15~16. 课程总结（1/2、2/2）

## 1.3 每周作业

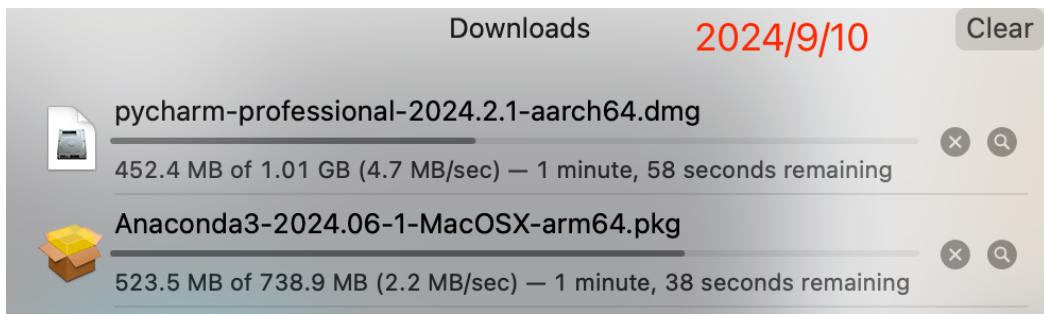
通常完成时限是一周。Assignment0, Assignment1 和 AssignmentP已经发布。

因为学校Canvas教学平台，课程补退选结束后9月19日才启动，前三周作业都是10月1日周二23:59截止。之后每次作业是下周二23:59截止。

### 1.3.1 集成开发环境

写python程序，通常是建议使用PyCharm (<https://www.jetbrains.com/pycharm/>)，anaconda/spyder (<https://www.anaconda.com/>)，这两个计算中心的机房有，考试的时候可以用。另外，机房也有vscode。

PyCharm Professional提供适用于学生和教师的免费许可，<https://www.jetbrains.com.cn/community/education/#students>



## 1.3.2 编程平台

我们的做题平台主要使用 OJ (<http://cs101.openjudge.cn>) 和 CF (<https://codeforces.com/>) 。

我们也欢迎使用 Luogu (<https://www.luogu.com.cn>) 和 LeetCode (<https://leetcode.cn>) 。

特别是 OJ，对于后续的《数据结构与算法B》课程来说非常重要，几乎所有的班级都在使用它。

[https://github.com/GMyhf/2024spring-cs201/blob/main/progress\\_of\\_various\\_classes.md](https://github.com/GMyhf/2024spring-cs201/blob/main/progress_of_various_classes.md)

题目ID, 标题, 描述

# CS101

加入

## »计概2024fall每日选做(10题)

最新题目

题目ID	标题	通过率	通过人数	尝试人数	添加时间
23564	数论	100%	5	5	2024-09-10
01003	Hangover	94%	30	32	2024-09-09
18223	24点	93%	26	28	2024-09-08
21532	数学密码	90%	27	30	2024-09-08
19949	提取实体v0.3 (string)	87%	20	23	2024-09-07

## »数算2024spring每日选做(156题)

最新题目

题目ID	标题	通过率	通过人数	尝试人数	添加时间
19930	寻宝	98%	47	48	2024-06-04
01798	Truck History	92%	22	24	2024-06-03
22460	火星车勘探	98%	47	48	2024-06-02
28389	跳高	90%	35	39	2024-06-01
02815	城堡问题	100%	20	20	2024-05-31

## »数算2024pre每日选做(102题)

最新题目

题目ID	标题	通过率	通过人数	尝试人数	添加时间
08758	2的幂次方表示	93%	27	29	2024-02-17
01611	The Suspects	93%	13	14	2024-02-17
24588	后序表达式求值	100%	32	32	2024-02-17
06250	字符串最大跨距	100%	14	14	2024-02-16
02337	Catenyms	56%	5	9	2024-02-16

## »计概2023fall每日选做(206题)

最新题目

题目ID	标题	通过率	通过人数	尝试人数	添加时间
04109	公共朋友–Common Friends	94%	16	17	2023-12-28
04112	情报破译–Cryptanalysis	100%	6	6	2023-12-28
01802	垂直直方图	96%	24	25	2023-12-27
07810	19岁生日礼物–Birthday Gift	100%	37	37	2023-12-27
21532	数学密码	91%	43	47	2023-12-26

## »题库 (包括计概、数算题目) (745题)

最新题目

题目ID	标题	通过率	通过人数	尝试人数	添加时间
15291	华容道	75%	3	4	2024-07-03
02528	Mayor's posters	100%	1	1	2024-07-03

图 <http://cs101.openjudge.cn> 平台

The screenshot shows the homepage of the <http://cs101.openjudge.cn> platform. At the top, there is a navigation bar with links: HOME, TOP, CATALOG, CONTESTS, GYM, PROBLEMSET, GROUPS, RATING, EDU, API, CALENDAR, and HELP. The PROBLEMSET link is underlined with a red wavy line. Below the navigation bar, there is a sub-navigation bar with links: MAIN, ACMMSGURU, PROBLEMS (which is highlighted in a grey box), SUBMIT, STATUS, STANDINGS, and CUSTOM TEST. The main content area is titled "Problems" and contains a table of problems. The table has columns for "#", "Name", "Description", "Solved by", "Rating", and "Handle". The table lists 12 problems, each with a green background and a small icon next to the handle.

#	Name	Description	Solved by	Rating	Handle
4A	<a href="#">Watermelon</a>	brute force, math		800	x512401
71A	<a href="#">Way Too Long Words</a>	strings		800	x375723
231A	<a href="#">Team</a>	brute force, greedy		800	x318417
1A	<a href="#">Theatre Square</a>	math		1000	x257362
282A	<a href="#">Bit++</a>	implementation		800	x255065
158A	<a href="#">Next Round</a>	*special problem, implementation		800	x244983
50A	<a href="#">Domino_piling</a>	greedy, math		800	x241762
263A	<a href="#">Beautiful Matrix</a>	implementation		800	x239790
112A	<a href="#">Petya and Strings</a>	implementation, strings		800	x221105
339A	<a href="#">Helpful Maths</a>	greedy, implementation, sortings, strings		800	x210251
281A	<a href="#">Word Capitalization</a>	implementation, strings		800	x207731

图 <https://codeforces.com/> 平台

## 1.4 开课前说明

### 1.4.1 练习盲打

[https://github.com/GMyhf/2024fall-cs101/blob/main/question1\\_before\\_class.md](https://github.com/GMyhf/2024fall-cs101/blob/main/question1_before_class.md)

### 1.4.2 查看题目测试数据、代码

1) 对于题目数据（通常编号介于1000至3256之间，例如：02692: 假币问题），当同学们有需求时，将会以单一的zip文件形式单独提供。请注意，出于管理规定，我们不允许批量发送题目测试数据。

2) 如果您希望查看特定同学提交的代码（如通过题目统计栏查看），无论是0毫秒的C++代码，还是21毫秒的Python3代码，都可以直接在群内提出请求。

The screenshot shows a competition interface for CS101. At the top, there's a navigation bar with icons for users, competition name, and tabs for '题目' (Questions), '排名' (Ranking), '状态' (Status), and '提问' (Ask). Below the navigation is a section titled '02692:假币问题' (Counterfeit Money Problem) with tabs for '查看' (View), '提交' (Submit), '统计' (Statistics), and '提问' (Ask). The main area displays two tables. The left table lists submissions from users 23n2300011079, 23n2300011079, 23n2300016304(世刚), 23n2300016304(世刚), 23n2300016304(世刚), and 23n2300016304(世刚). The right table shows the distribution of errors: Wrong Answer (714), Accepted (401), Runtime Error (136), Compile Error (67), and Presentation Error (4).

提交人	结果	内存	时间	代码长度	语言	提交时间
23n2300011079	Accepted	248kB	0ms	1061 B	C++	24.2.22
23n2300011079	Compile Error			1060 B	C++	24.2.22
23n2300016304(世刚)	Accepted	3616kB	21ms	849 B	Python3	23.12.27
23n2300016304(世刚)	Wrong Answer	3612kB	21ms	849 B	Python3	23.12.27
23n2300016304(世刚)	Wrong Answer	3560kB	18ms	851 B	Python3	23.12.27
23n2300016304(世刚)	Runtime Error	3544kB	18ms	686 B	Python3	23.12.27

统计	结果	提交次数
Wrong Answer	714	
Accepted	401	
Runtime Error	136	
Compile Error	67	
Presentation Error	4	

### 1.4.3 提交作业使用软件

- 1) 我们班的计概及后续课程将会使用 Markdown 文件，扩展名是md。Typora 是一款优秀的 Markdown 编辑软件，但学校并未提供免费版本。
- 2) 大家可以关注一下目前的一个优惠活动（距离活动结束还有7天），优惠价格为80.10元。具体信息可参考以下链接：Typora官网 <https://typoraio.cn>，优惠购买页面 <https://lizhi.shop/site/products/id/520>
- 3) 过去也有三位同学一起购买的情况，因为一个许可证可以激活三台设备。
- 4) 购买并非强制性的。以往也有同学使用 Word 或其他软件完成作业，同样没有问题。

typora是文本编辑器，可以写markdown格式文件，课件、作业用，python程序加入markdown中不乱的。pycharm, anaconda/spyder是写python程序的集成开发环境，选择一个就可以。

听闻有Typora序列号生成机器，我有正版，没有尝试。<https://blog.csdn.net/niceBCZZ/article/details/137568047>，亲测可用

- 5) 打开扩展是pdf的软件，如：<http://software.pku.edu.cn> 北大提供 福昕PDF编辑软件。

### 1.4.4 选课相关

树洞，<https://treehole.pku.edu.cn/web/>

课程评测@PKU，<https://courses.pinzhixiaoyuan.com>

公众号：赛艇先生

## 20240906

- 1) 访问选课系统 (<elective.pku.edu.cn>)，看到教务调整到开课人数150（之前是120），已选人数150。

2) 不知道后期是否还调整（同学可以邮件等方式联系 信息科学技术学院教务：董晓晖，地点：理科1号楼1118，电话：010-62755414，邮箱：[dxh@pku.edu.cn](mailto:dxh@pku.edu.cn)），教务老师昨天回复我是“统筹安排”。

3) 即使没有选上我们班的同学，如果愿意，也可以一起探讨编程问题。

课程微信群，在里面C, C++, Python问题都可以问的。课程资料也都会发的。

入门语法：

<https://www.runoob.com/python3/python3-tutorial.html>

<https://www.runoob.com/cprogramming/c-tutorial.html>

<https://www.runoob.com/cplusplus/cpp-tutorial.html>

语法掌握后，就是在 CF, OJ 做题：

我们主要是用OJ平台，后续课程数算也是这平台。

<http://cs101.openjudge.cn>，另外我们英文题目用 <https://codeforces.com>。这两个平台大家都需要注册账号。

编程语言是想通的，掌握一个语法后，很快就会另外一个，不会冲突的。

## 20240904

1) 根据往年情况，选课时候，如果学生找教务多，教务会慢慢放出名额，尽量让同学选上。我这里没有选课权限，加不了学生。

信息科学技术学院教务：董晓晖，地点：理科1号楼1118，电话：010-62755414，邮箱：[dxh@pku.edu.cn](mailto:dxh@pku.edu.cn)

2) 有经验的教务老师，根据每年选课人数，会有调整策略。例如：目前（2024年5月10日）校本部计算概论（B）每个班选课人数先录入150人，是想要大点的教室，避免教室过于拥挤。待教室排定后，我再把人数降到120人。

## 20240829

我们班上课地点在二教107，有217个座位；上机地点7号机房学生机109台，8号102台。

选课权限在教务老师（目前20240906看到是初设的120开课人数，提高到150，再提高到180），我这里不过滤都接收，需要签字同意等都没有问题。往年教务老师会根据选课情况逐渐调整（第一周放点名额，第二周放点名额），比如调整到150或者更高到教室容量的上限。

任课教师和助教无权帮助学生选课。信息科学技术学院教务：董晓晖，地点：理科1号楼1118，电话：010-62755414，邮箱：[dxh@pku.edu.cn](mailto:dxh@pku.edu.cn)

## 20240826

同学选课后，还要自动抽签。之后权限在教务老师，我这里不过滤都接收，需要签字同意等都没有问题。往年教务老师会根据选课情况调整，比如调整到教室容量的上限。

信息科学技术学院教务：董晓晖，地点：理科1号楼1118，电话：010-62755414，邮箱：[dxh@pku.edu.cn](mailto:dxh@pku.edu.cn)

权限都在教务！第0周（开课前一周）周五对超过预定选课人数的课程进行抽签，公布选课结果；之后安排课程补选、退选，学生可以在网上查看自己的已选课程记录，并可退课或补选。其中：第一周周三晚各院系本科必修和限选课程的剩余名额向全校所有本科生开放选课；系统给各院系教务预留时间设置课程对外开放名额，期间学生不能选课；第二周周二上午10: 00—周四上午10: 00，只能补选，不能退课；此后选课系统关闭，网上选课结束。

## 1.4.5 上机安排和零基础启动

1) 一直以来“上机课”0学分，都不录入成绩。可以不选的。但是期末机考一定要参加，是上机时间。  
另外有每个月初有一次月考，是在机房进行，尽量参加，熟悉环境。

上机课是助教答疑时间，同学自行安排是否出现。最后一次上机（2025.12.26周四 7-8节）考试出现是必须的。

2) 零基础同学尽快熟悉入门语法：

<https://www.runoob.com/python3/python3-tutorial.html>

<https://www.runoob.com/cprogramming/c-tutorial.html>

<https://www.runoob.com/cplusplus/cpp-tutorial.html>

这些内容将帮助学习者掌握基本的语法结构和概念，为后续编程学习打下基础。

语法掌握后，就是在 CF，OJ 做题：我们主要是用 OJ 平台，后续课程数算也是这平台。

<http://cs101.openjudge.cn>，另外我们英文题目用 <https://codeforces.com>。这两个平台大家都需要注册账号。

编程语言是想通的，掌握一个语法后，很快就会另外一个，不会冲突的。

3) 跟上每日选做，开始放出来的都是最简单的题目。题目以及之后的作业，都有题解同时发布。

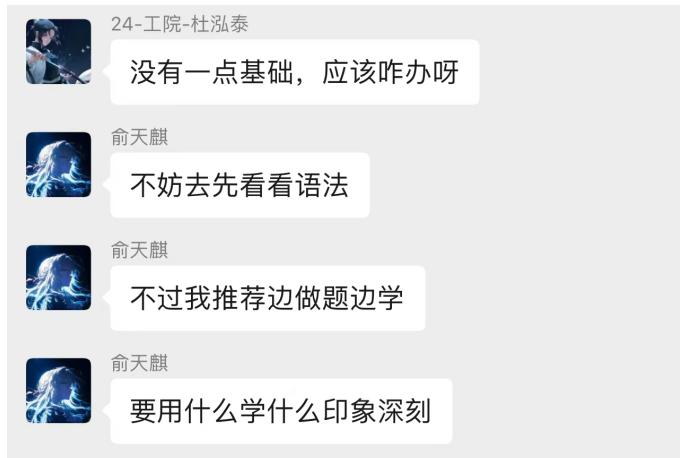
4) 请善用gpt。poe.com需要翻墙。国内阿里的 通义千问也好用，<https://tongyi.aliyun.com>。课程相关问题，都可以在课程微信群里面问。例如：考试时间、作业、上机、签到、大作业有否等。

5) 可能需要北京大学长提供的Clash，请自己取用。

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

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

任何相关问题，都可以在大群问。私人问题可以问我或者助教。





## 1.5 其他说明

### 1.5.1 课程微信群

请同学修改课程微信群昵称为实名。

救助在自己姓名前面有 "TA-", 如: TA-熊江凯, TA-罗熙佑, TA-涂程颖, TA-王嘉林, TA-刘昊文

同学互助答疑

## 2024fall-cs101: Algo DS (415)



24-物院-曹以楷

编译器的话可以在 code runner 的 executormap 中，把编译命令中手动加上 -finput-charset=UTF-8 -fexec-charset=UTF-8，来迫使 g++ 编译的时候用 utf-8



24-物院-曹以楷

system("chcp 65001") 这个是用来设置终端的编码的



化院萧博凡

(已选择21) 空格: 4 UTF-8 CRLF {}



化院萧博凡

我好像就是 utf-8

2024/9/9

### 1.5.2 学生成绩评定方法上

第16周，来机考就可以。

我们采用综合评定方法对学生成绩进行评定。学生成绩分为三部分，大致占比是30%的成绩来自每周1次的作业（其中有3次作业对应3次月考）成绩，另外70%的成绩来自期末上机考试、笔试。我们注重学期中的学习过程评定，因此成绩评定不仅考虑学生的考试成绩，还综合考虑学生的平时表现和参与度。

10月、11月、12月，每个月初有一次月考（建议尽量在机房规定时间完成），月考题目也会留作当周作业。因为有月考，不安排期中考试。月考不计分，没有监考。都是周四上机时间，分别是 10月10日，11月7日，12月5日。

### 1.5.3 Canvas课程网站

教发中心通知，由于Canvas付费用户帐号数量有限（每全年仅支持3000多个师生用户），新学期Canvas用户数量将严格控制，学生名单将在补退选结束后导入（本科生9月19日，研究生9月24日）。特殊情况，请发邮件说明。

O365（含teams）账号申请链接：<https://www.wjx.cn/vm/Y5XwfHD.aspx#>

Canvas 是学校为我们班级提供的平台，选课的同学可以免费使用。O365 是面向全校的服务，即使没有选课也可以申请使用。

## 二、导论

### 1.1 入门

自从第一台利用转接线和开关来传递计算指令的电子计算机诞生以来，人们对编程的认识历经了多次变化。与社会生活的其他许多方面一样，计算机技术的变革为计算机科学家提供了越来越多的工具和平台去施展他们的才能。高效的处理器、高速网络以及大容量内存等一系列新技术，要求计算机科学家掌握更多复杂的知识。然而，在这一系列快速的变革之中，仍有一些基本原则始终保持不变。计算机科学被认为是一门利用计算机来解决问题的学科。

用 <https://browser.geekbench.com/> 测试计算设备性能。

The screenshot shows the Geekbench browser interface. At the top, there are tabs for Geekbench 6 (4), Geekbench 5 (14), Geekbench 4 (0), Geekbench 3 (0), and Geekbench 2 (0). To the right, the URL [browser.geekbench.com](https://browser.geekbench.com/) is displayed. Below the tabs, there are two buttons: 'CPU' (2) and 'Compute' (2). A table lists two devices:

ID	Name	Platform	Architecture	Single-core Score	Multi-core Score
972051	iPhone 14 Pro Max Apple A16 Bionic 3460 MHz (6 cores)	iOS	AArch64	2528	6352
971930	Mac Studio Apple M1 Ultra 3219 MHz (20 cores)	macOS	AArch64	2377	18048

图 我的在用设备性能

browser.geekbench.com/user/433953

The screenshot shows the Geekbench Browser interface with the following details:

- Header:** Geekbench Browser, Geekbench 6, Geekbench AI, Benchmark Charts, Search, depot.
- Navigation:** Geekbench 6 (4), Geekbench 5 (14), Geekbench 4 (0), Geekbench 3 (0), Geekbench 2 (0).
- Filter:** CPU (14) selected, Compute (0).
- Table Headers:** ID, Name, Platform, Architecture, Single-core Score, Multi-core Score.
- Table Data:** A list of 15 devices with their scores and specifications, including Mac mini (Late 2014), Apple Inc. Macmini7,1, iPhone XS Max, D Precision 5820 Tower, iPhone 6 Plus, iPhone XS Max, iPhone 11 Pro Max, iPhone 5s, iPhone 6s, iPhone 12 Pro Max, MacBook Air (13-inch Early 2015), MacBook Pro (15-inch Retina Mid 2015), MacBook Pro (13-inch Retina Early 2015), and Mac mini (Late 2020).

ID	Name	Platform	Architecture	Single-core Score	Multi-core Score
16311392	Mac mini (Late 2014) Intel Core i5-4278U 2600 MHz (2 cores)	macOS	x64	757	1653
16186750	Apple Inc. Macmini7,1 Intel Core i5-4308U 2799 MHz (2 cores)	Windows	x64	686	1556
15965721	iPhone XS Max Apple A12 Bionic 2490 MHz (6 cores)	iOS	AArch64	1106	2295
15965487	D Precision 5820 Tower Intel Xeon W-2223 3592 MHz (4 cores)	Windows	x64	643	3963
15965077	iPhone 6 Plus Apple A8 1400 MHz (2 cores)	iOS	AArch64	337	607
15965076	iPhone XS Max Apple A12 Bionic 2490 MHz (6 cores)	iOS	AArch64	1115	2817
15950459	iPhone 11 Pro Max Apple A13 Bionic 2660 MHz (6 cores)	iOS	AArch64	1328	3442
15950414	iPhone 5s Apple A7 1300 MHz (2 cores)	iOS	AArch64	272	452
15950323	iPhone 6s Apple A9 1849 MHz (2 cores)	iOS	AArch64	555	1026
15926386	iPhone 12 Pro Max Apple A14 Bionic 2990 MHz (6 cores)	iOS	AArch64	1595	4153
15915212	MacBook Air (13-inch Early 2015) Intel Core i5-5250U 1600 MHz (2 cores)	macOS	x64	511	1014
15914716	MacBook Pro (15-inch Retina Mid 2015) Intel Core i7-4870HQ 2500 MHz (4 cores)	macOS	x64	817	3312
15914652	MacBook Pro (13-inch Retina Early 2015) Intel Core i5-5287U 2900 MHz (2 cores)	macOS	x64	690	1599
15914307	Mac mini (Late 2020) Apple M1 3197 MHz (8 cores)	macOS	AArch64	1735	7556

图 以往设备性能

用Speedtest测试网络速度。



图 2024/9/10 理科1号楼有线网络

## 1.2 何谓计算机科学

要定义计算机科学，通常十分困难，这也许是因为其中的“计算机”一词。你可能已经意识到，计算机科学并不仅是研究计算机本身。尽管计算机在这一学科中是非常重要的工具，但也仅仅只是工具而已。

计算机科学的研究对象是问题、解决问题的过程，以及通过该过程得到的解决方案。给定一个问题，计算机科学家的目标是开发一个能够逐步解决该问题的**算法**。算法是具有有限步骤的过程，依照这个过程便能解决问题。因此，**算法**就是解决方案。

可以认为计算机科学就是研究算法的学科。但是必须注意，某些问题并没有解决方案。尽管这一话题已经超出了本书讨论的范畴，但是对于学习计算机科学的人来说，认清这一事实非常重要。结合上述两类问题，可以将计算机科学更完善地定义为：研究问题及其解决方案，以及研究目前无解的问题的学科。

在描述问题及其解决方案时，经常用到“可计算”一词。若存在能够解决某个问题的算法，那么该问题便是可计算的。因此，计算机科学也可以被定义为：研究可计算以及不可计算的问题，即研究算法的存在性以及不存在性。在上述任意一种定义中，“计算机”一词都没有出现。解决方案本身是独立于计算机的。

在研究问题解决过程的同时，计算机科学也研究**抽象**。抽象思维使得我们能分别从逻辑视角和物理视角来看待问题及其解决方案。举一个常见的例子。

试想你每天开车去上学或上班。作为车的使用者，你在驾驶时会与它有一系列的交互：坐进车里，插入钥匙，启动发动机，换挡，刹车，加速以及操作方向盘。从抽象的角度来看，这是从逻辑视角来看待这辆车，你在使用由汽车设计者提供的功能来将自己从某个地方运送到另一个地方。这些功能有时候也被称作**接口**。

另一方面，修车工看待车辆的角度与司机截然不同。他不仅需要知道如何驾驶，而且更需要知道实现汽车功能的所有细节：发动机如何工作，变速器如何换挡，如何控制温度，等等。这就是所谓的物理视角，即看到表面之下的实现细节。

使用计算机也是如此。大多数人用计算机来写文档、收发邮件、浏览网页、听音乐、存储图像以及打游戏，但他们并不需要了解这些功能的实现细节。大家都是从逻辑视角或者使用者的角度来看待计算机。计算机科学家、程序员、技术支持人员以及系统管理员则从另一个角度来看待计算机。他们必须知道操作系统的原理、网络协议的配置，以及如何编写各种脚本来控制计算机。他们必须能够控制用户不需要了解的底层细节。

上面两个例子的共同点在于，抽象的用户（或称客户）只需要知道接口是如何工作的，而并不需要知道实现细节。这些接口是用户用于与底层复杂的实现进行交互的方式。下面是抽象的另一个例子，来看看Python的math模块。一旦导入这一模块，便可以进行如下的计算。

```
1 | >>> import math  
2 | >>> math.sqrt(16)  
3 | 4.0  
4 | >>>
```

这是一个过程抽象的例子。我们并不需要知道平方根究竟是如何计算出来的，而只需要知道计算平方根的函数名是什么以及如何使用它。只要正确地导入模块，便可以认为这个函数会返回正确的结果。由于其他人已经实现了平方根问题的解决方案，因此我们只需要知道如何使用该函数即可。这有时候也被称为过程的“黑盒”视角。我们仅需要描述接口：函数名、所需参数，以及返回值。所有的计算细节都被隐藏了起来，如图1-1所示。



图1-1 过程抽象

### 1.3.1 何谓编程

编程是指通过编程语言将算法编码以使其能被计算机执行的过程。尽管有众多的编程语言和不同类型的计算机，但是首先得有一个解决问题的算法。如果没有算法，就不会有程序。

计算机科学的研究对象并不是编程。但是，编程是计算机科学家所做工作的一个重要组成部分。通常，编程就是为解决方案创造表达方式。因此，编程语言对算法的表达以及创造程序的过程是这一学科的基础。

通过定义表达问题实例所需的数据，以及得到预期结果所需的计算步骤，算法描述出了问题的解决方案。编程语言必须提供一种标记方式，用于表达过程和数据。为此，编程语言提供了众多的控制语句和数据类型。

Programming is often the way that we create a representation for our solutions. Therefore, this language representation and the process of creating it becomes a fundamental part of the discipline.

控制语句使算法步骤能够以一种方便且明确的方式表达出来。算法至少需要能够进行顺序执行、决策分支、循环迭代的控制语句。只要一种编程语言能够提供这些基本的控制语句，它就能够被用于描述算法。

计算机中的所有数据实例均由二进制字符串来表达。为了赋予这些数据实际的意义，必须要有数据类型。数据类型能够帮助我们解读二进制数据的含义，从而使我们能从待解决问题的角度来看待数据。这些内建的底层数据类型（又称原生数据类型）提供了算法开发的基本单元。

举例来说，大部分编程语言都为整数提供了相应的数据类型。根据整数（如23、654以及-19）的常见定义，计算机内存中的二进制字符串可以被理解成整数。除此以外，数据类型也描述了该类数据能参与的所有运算。对于整数来说，就有加减乘除等常见运算。并且，对于数值类型的数据，以上运算均成立。

我们经常遇到的困难是，问题及其解决方案都过于复杂。尽管由编程语言提供的简单的控制语句和数据类型能够表达复杂的解决方案，但它们在解决问题的过程中仍然存在不足。因此，我们需要想办法控制复杂度以利于找到解决方案。

### 1.3.2 为何学习算法

计算机科学家通过经验来学习：观察他人如何解决问题，然后亲自解决问题。接触各种问题解决技巧并学习不同算法的设计方法，有助于解决新的问题。通过学习一系列不同的算法，可以举一反三，从而在遇到类似的问题时，能够快速加以解决。

各种算法之间往往差异巨大。回想前文提到的平方根的例子，完全可能有多种方法来实现计算平方根的函数。算法一可能使用了较少的资源，算法二返回结果所需的时间可能是算法一的10倍。我们需要某种方式来比较这两种算法。尽管这两种算法都能得到结果，但是其中一种可能比另一种“更好”——更高效、更快，或者使用的内存更少。随着对算法的进一步学习，你会掌握比较不同算法的分析技巧。这些技巧只依赖于算法本身的特性，而不依赖于程序或者实现算法的计算机的特性。

最坏的情况是遇到难以解决的问题，即没有算法能够在合理的时间内解决该问题。因此，至关重要的一点是，要能区分有解的问题、无解的问题，以及虽然有解但是需要过多的资源和时间来求解的问题。

在选择算法时，经常会有所权衡。除了有解决问题的能力之外，计算机科学家也需要知晓如何评估一个解决方案。总之，问题通常有很多解决方案，如何找到一个解决方案并且确定其为优秀的方案，是需要反复练习、熟能生巧的。

## 1.4 Python基础

本节将为前一节提到的思想提供更详细的例子。

Python是一门现代、易学、面向对象的编程语言。它拥有强大的内建数据类型以及简单易用的控制语句。由于Python是一门解释型语言，因此只需要查看和描述交互式会话就能进行学习。你应该记得，解释器会显示提示符>>>，然后计算你提供的Python语句。例如，以下代码显示了提示符、print函数、结果，以及下一个提示符。

```
1 | >>> print("Algorithms and Data Structures")
2 | >>> Algorithms and Data Structures
3 | >>>
```

### 1.4.1 数据

前面提到，Python支持面向对象编程范式。这意味着Python认为数据是问题解决过程中的关键点。在Python以及其他所有面向对象编程语言中，类都是对数据的构成（状态）以及数据能做什么（行为）的描述。由于类的使用者只能看到数据项的状态和行为，因此类与抽象数据类型是相似的。在面向对象编程范式中，数据项被称作对象。一个对象就是类的一个实例。

## 1. 内建原子数据类型

我们首先复习原子数据类型。Python有两大内建数据类实现了整数类型和浮点数类型，相应的Python类就是int和float。标准的数学运算符，即+、-、\*、/以及\*\*（幂），可以和能够改变运算优先级的括号一起使用。其他非常有用的运算符包括取余（取模）运算符%，以及整除运算符//。注意，当两个整数相除时，其结果是一个浮点数，而整除运算符截去小数部分，只返回商的整数部分。

Python通过bool类实现对表达真值非常有用的布尔数据类型。布尔对象可能的状态值是True或者False，布尔运算符有and、or以及not。

布尔对象也被用作相等（==）、大于（>）等比较运算符的计算结果。此外，结合使用关系运算符与逻辑运算符可以表达复杂的逻辑问题。表1-1展示了关系运算符和逻辑运算符。

表1-1 关系运算符和逻辑运算符

Operation Name	Operator	Explanation
less than	<	Less than operator
greater than	>	Greater than operator
less than or equal	<=	Less than or equal to operator
greater than or equal	>=	Greater than or equal to operator
equal	==	Equality operator
not equal	!=	Not equal operator
logical and	and	Both operands True for result to be True
logical or	or	One or the other operand is True for the result to be True
logical not	not	Negates the truth value, False becomes True, True becomes False

标识符在编程语言中被用作名字。Python中的标识符以字母或者下划线（\_）开头，区分大小写，可以是任意长度。需要记住的一点是，采用能表达含义的名字是良好的编程习惯，这使程序代码更易阅读和理解。

当一个名字第一次出现在赋值语句的左边部分时，会创建对应的Python变量。赋值语句将名字与值关联起来。变量存的是指向数据的引用，而不是数据本身。

赋值语句改变了变量的引用，这体现了Python的动态特性。同样的变量可以指向许多不同类型的数据。

## 2. 内建集合数据类型

除了数值类和布尔类，Python还有众多强大的内建集合类。列表、字符串以及元组是概念上非常相似的有序集合，但是只有理解它们的差别，才能正确运用。集（set）和字典是无序集合。

**列表**是零个或多个指向Python数据对象的引用的有序集合，通过在方括号内以逗号分隔的一系列值来表达。空列表就是[]。列表是异构的，这意味着其指向的数据对象不需要都是同一个类，并且这一集合可以被赋值给一个变量。

由于列表是有序的，因此它支持一系列可应用于任意Python序列的运算，如表1-2所示。

表1-2 可应用于任意Python序列的运算

Operation Name	Operator	Explanation
indexing	[ ]	Access an element of a sequence
concatenation	+	Combine sequences together
repetition	*	Concatenate a repeated number of times
membership	in	Ask whether an item is in a sequence
length	len	Ask the number of items in the sequence
slicing	[ : ]	Extract a part of a sequence

需要注意的是，列表和序列的下标从0开始。myList[1:3]会返回一个包含下标从1到2的元素列表（并没有包含下标为3的元素）。

如果需要快速初始化列表，可以通过重复运算来实现，如下所示。

```
1 >>> myList = [0] * 6
2 >>> myList
3 [0, 0, 0, 0, 0, 0]
```

非常重要的一点是，重复运算返回的结果是序列中指向数据对象的引用的重复。下面的例子可以很好地说明这一点。

```
1 >>> myList = [1,2,3,4]
2 >>> A = [myList]*3
3 >>> A
4 [[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]
5 >>> myList[2] = 45
6 >>> A
7 [[1, 2, 45, 4], [1, 2, 45, 4], [1, 2, 45, 4]]
```

变量A包含3个指向myList的引用。myList中的一个元素发生改变，A中的3处都随即改变。

列表支持一些用于构建数据结构的方法，如表1-3所示。后面的例子展示了用法。

表1-3 Python列表提供的方法

Method Name	Use	Explanation
append	<code>alist.append(item)</code>	Adds a new item to the end of a list
insert	<code>alist.insert(i,item)</code>	Inserts an item at the ith position in a list
pop	<code>alist.pop()</code>	Removes and returns the last item in a list
pop	<code>alist.pop(i)</code>	Removes and returns the ith item in a list
sort	<code>alist.sort()</code>	Modifies a list to be sorted
reverse	<code>alist.reverse()</code>	Modifies a list to be in reverse order
del	<code>del alist[i]</code>	Deletes the item in the ith position
index	<code>alist.index(item)</code>	Returns the index of the first occurrence of <code>item</code>
count	<code>alist.count(item)</code>	Returns the number of occurrences of <code>item</code>
remove	<code>alist.remove(item)</code>	Removes the first occurrence of <code>item</code>

你会发现，像pop这样的方法在返回值的同时也会修改列表的内容，reverse等方法则仅修改列表而不返回任何值。pop默认返回并删除列表的最后一个元素，但是也可以用来返回并删除特定的元素。这些方法默认下标从0开始。你也会注意到那个熟悉的句点符号，它被用来调用某个对象的方法。

range是一个常见的Python函数，我们常把它与列表放在一起讨论。range会生成一个代表值序列的范围对象。使用list函数，能够以列表形式看到范围对象的值。

范围对象表示整数序列。默认情况下，它从0开始。如果提供更多的参数，它可以在特定的点开始和结束，并且跳过中间的值。在第一个例子中，`range(10)`从0开始并且一直到9为止（不包含10）；在第二个例子中，`range(5,10)`从5开始并且到9为止（不包含10）；`range(5,10,2)`的结果类似，但是元素的间隔变成了2（10还是没有包含在其中）。

字符串是零个或多个字母、数字和其他符号的有序集合。这些字母、数字和其他符号被称为字符。常量字符串值通过引号（单引号或者双引号均可）与标识符进行区分。

由于字符串是序列，因此之前提到的所有序列运算符都能用于字符串。此外，字符串还有一些特有的方法，表1-4列举了其中一些。

表1-4 Python字符串提供的方法

Method Name	Use	Explanation
center	astring.center(w)	Returns a string centered in a field of size w
count	astring.count(item)	Returns the number of occurrences of item in the string
ljust	astring.ljust(w)	Returns a string left-justified in a field of size w
lower	astring.lower()	Returns a string in all lowercase
rjust	astring.rjust(w)	Returns a string right-justified in a field of size w
find	astring.find(item)	Returns the index of the first occurrence of item
split	astring.split(schar)	Splits a string into substrings at schar

split在处理数据的时候非常有用。split接受一个字符串，并且返回一个由分隔字符作为分割点的字符串列表。

列表和字符串的主要区别在于，列表能够被修改，字符串则不能。列表的这一特性被称为可修改性。列表具有可修改性，字符串则不具有。

由于都是异构数据序列，因此元组与列表非常相似。它们的区别在于，元组和字符串一样是不可修改的。元组通常写成由括号包含并且以逗号分隔的一系列值。与序列一样，元组允许之前描述的任一操作。

集 (set) 是由零个或多个不可修改的Python数据对象组成的无序集合。集不允许重复元素，并且写成由花括号包含、以逗号分隔的一系列值。空集由set()来表示。集是异构的。

尽管集是无序的，但它还是支持之前提到的一些运算，如表1-5所示。

表1-5 Python集支持的运算

Operation Name	Operator	Explanation
membership	in	Set membership
length	len	Returns the cardinality of the set
	aset   otherset	Returns a new set with all elements from both sets
&	aset & otherset	Returns a new set with only those elements common to both sets
-	aset - otherset	Returns a new set with all items from the first set not in second
<=	aset <= otherset	Asks whether all elements of the first set are in the second

集支持一系列方法，如表1-6所示。在数学中运用过集合概念的人应该对它们非常熟悉。Note that `union`, `intersection`, `issubset`, and `difference` all have operators that can be used as well.

Note that `union`, `intersection`, `issubset`, and `difference` all have operators that can be used as well.

表1-6 Python集提供的方法

Method Name	Use	Explanation
<code>union</code>	<code>aset.union(otherset)</code>	Returns a new set with all elements from both sets
<code>intersection</code>	<code>aset.intersection(otherset)</code>	Returns a new set with only those elements common to both sets
<code>difference</code>	<code>aset.difference(otherset)</code>	Returns a new set with all items from first set not in second
<code>issubset</code>	<code>aset.issubset(otherset)</code>	Asks whether all elements of one set are in the other
<code>add</code>	<code>aset.add(item)</code>	Adds item to the set
<code>remove</code>	<code>aset.remove(item)</code>	Removes item from the set
<code>pop</code>	<code>aset.pop()</code>	Removes an arbitrary element from the set
<code>clear</code>	<code>aset.clear()</code>	Removes all elements from the set

字典是无序结构，由相关的元素对构成，其中每对元素都由一个键和一个值组成。这种键-值对通常写成键：值的形式。字典由花括号包含的一系列以逗号分隔的键-值对表达，

可以通过键访问其对应的值，也可以向字典添加新的键-值对。访问字典的语法与访问序列的语法十分相似，只不过是使用键来访问，而不是下标。添加新值也类似。

需要谨记，字典并不是根据键来进行有序维护的。键的位置是由散列来决定的，后续章节会详细介绍散列。`len`函数对字典的功能与对其他集合的功能相同。

字典既有运算符，又有方法。表1-7和表1-8分别展示了它们。`keys`、`values`和`items`方法均会返回包含相应值的对象。可以使用`list`函数将字典转换成列表。在表1-8中可以看到，`get`方法有两种版本。如果键没有出现在字典中，`get`会返回`None`。然而，第二个可选参数可以返回特定值。

表1-7 Python字典支持的运算

Operator	Use	Explanation
<code>[]</code>	<code>myDict[k]</code>	Returns the value associated with <code>k</code> , otherwise its an error
<code>in</code>	<code>key in adict</code>	Returns <code>True</code> if key is in the dictionary, <code>False</code> otherwise
<code>del</code>	<code>del adict[key]</code>	Removes the entry from the dictionary

表1-8 Python字典提供的方法

Method Name	Use	Explanation
keys	<code>adict.keys()</code>	Returns the keys of the dictionary in a dict_keys object
values	<code>adict.values()</code>	Returns the values of the dictionary in a dict_values object
items	<code>adict.items()</code>	Returns the key-value pairs in a dict_items object
get	<code>adict.get(k)</code>	Returns the value associated with <code>k</code> , <code>None</code> otherwise
get	<code>adict.get(k, alt)</code>	Returns the value associated with <code>k</code> , <code>alt</code> otherwise

## 1.4.2 输入与输出

程序经常需要与用户进行交互，以获得数据或者提供某种结果。目前的大多数程序使用对话框作为要求用户提供某种输入的方式。尽管Python确实有方法来创建这样的对话框，但是可以利用更简单的函数。Python提供了一个函数，它使得我们可以要求用户输入数据并且返回一个字符串的引用。这个函数就是input。

input函数接受一个字符串作为参数。由于该字符串包含有用的文本来提示用户输入，因此它经常被称为提示字符串。

不论用户在提示字符串后面输入什么内容，都会被存储在aName变量中。使用input函数，可以非常简便地写出程序，让用户输入数据，然后再对这些数据进行进一步处理。

需要注意的是，input函数返回的值是一个字符串，它包含用户在提示字符串后面输入的所有字符。如果需要将这个字符串转换成其他类型，必须明确地提供类型转换。

### 格式化字符串

print函数为输出Python程序的值提供了一种非常简便的方法。它接受零个或者多个参数，并且将单个空格作为默认分隔符来显示结果。通过设置sep这一实际参数可以改变分隔符。此外，每一次打印都默认以换行符结尾。这一行为可以通过设置实际参数end来更改。

更多地控制程序的输出格式经常十分有用。幸运的是，Python提供了另一种叫作格式化字符串的方式。格式化字符串是一个模板，其中包含保持不变的单词或空格，以及之后插入的变量的占位符。

## 1.4.3 控制结构

算法需要两个重要的控制结构：迭代和分支。Python通过多种方式支持这两种控制结构。程序员可以根据需要选择最有效的结构。

对于迭代，Python提供了标准的while语句以及非常强大的for语句。while语句会在给定条件为真时重复执行一段代码。

分支语句允许程序员进行询问，然后根据结果，采取不同的行动。绝大多数的编程语言都提供两种有用的分支结构：if else和if。

和其他所有控制结构一样，分支结构支持嵌套，一个问题的结果能帮助决定是否需要继续问下一个问题。

另一种表达嵌套分支的语法是使用`elif`关键字。将`else`和下一个`if`结合起来，可以减少额外的嵌套层次。注意，最后的`else`仍然是必需的，它用来在所有分支条件都不满足的情况下提供默认分支。

列表可以不通过迭代结构和分支结构来创建，这种方式被称为列表解析式。通过列表解析式，可以根据一些处理和分支标准轻松创建列表。Returning to lists, there is an alternative method for creating a list that uses iteration and selection constructs known as a **list comprehension**. A list comprehension allows you to easily create a list based on some processing or selection criteria.

## 1.4.4 异常处理

在编写程序时通常会遇到两种错误。第一种是语法错误，也就是说，程序员在编写语句或者表达式时出错。

第二种是逻辑错误，即程序能执行完成但返回了错误的结果。这可能是由于算法本身有错，或者程序员没有正确地实现算法。有时，逻辑错误会导致诸如除以0、越界访问列表等非常严重的情况。这些逻辑错误会导致运行时错误，进而导致程序终止运行。通常，这些运行时错误被称为异常。

许多初级程序员简单地把异常等同于引起程序终止的严重运行时错误。然而，大多数编程语言都提供了让程序员能够处理这些错误的方法。此外，程序员也可以在检测到程序执行有问题的情况下自己创建异常。

当异常发生时，我们称程序“抛出”异常。可以用`try`语句来“处理”被抛出的异常。

## 1.4.5 定义函数

之前的过程抽象例子调用了Python数学模块中的`sqrt`函数来计算平方根。通常来说，可以通过定义函数来隐藏任何计算的细节。函数的定义需要一个函数名、一系列参数以及一个函数体。函数也可以显式地返回一个值。

## 1.5 Summary

---

- Computer science is the study of problem solving.
- Python is a powerful, yet easy-to-use, object-oriented language.
- Lists, tuples, and strings are built in Python sequential collections.
- Dictionaries and sets are nonsequential collections of data.

# 三、每日选做

2024fall 每日选做, [https://github.com/GMyhf/2024fall-cs101/blob/main/problem\\_list\\_at\\_2024fall.md](https://github.com/GMyhf/2024fall-cs101/blob/main/problem_list_at_2024fall.md)

从2024年8月23日开始, 每天都会增加2个题目供同学们自愿练习, 期中开始, 每天增加1个题目。争取每20个题目难度提升100。

建议大家跟上练习, 其中很多题目可能会是之后作业。此外, 机考在整个评估中占比较大, 保持每天编程的手感很重要。只有通过持续的努力和实践, 才能在机考中取得优异的成绩!

OJ上面太简单题目少, 所以开始尽量选了CF题目。加油! 感觉900难度有明显提升, 争取第一次课我们上习题课!

题解在 <https://github.com/GMyhf/2020fall-cs101> 中的 [2020fall\\_cs101.openjudge.cn\\_problems.md](#), [2020fall\\_Codeforces\\_problems.md](#)

The screenshot shows a browser window with the URL [https://github.com/GMyhf/2024fall-cs101/blob/main/problem\\_list\\_at\\_2024fall.md](https://github.com/GMyhf/2024fall-cs101/blob/main/problem_list_at_2024fall.md). The page content is as follows:

说明: 从2024年8月23日开始, 每天都会增加2个题目供同学们自愿练习, 期中开始, 每天增加1个题目。争取每20个题目难度提升100。

建议大家跟上练习, 其中很多题目可能会是之后作业。此外, 机考在整个评估中占比较大, 保持每天编程的手感很重要。只有通过持续的努力和实践, 才能在机考中取得优异的成绩!

OJ上面太简单题目少, 所以开始尽量选了CF题目。加油! 感觉900难度有明显提升, 争取第一次课我们上习题课!

题解在 <https://github.com/GMyhf/2020fall-cs101> 中的 [2020fall\\_cs101.openjudge.cn\\_problems.md](#), [2020fall\\_Codeforces\\_problems.md](#)

### 2024fall 每日选做

日期	问题编号与名称	标签	难度	链接
0911	03143: 验证哥德巴赫猜想	-	-	<a href="http://cs101.openjudge.cn/practice/03143">http://cs101.openjudge.cn/practice/03143</a>
	474A. Keyboard	implementation	900	<a href="https://codeforces.com/contest/474/problem/A">https://codeforces.com/contest/474/problem/A</a>
0910	23564: 数论	-	-	<a href="http://cs101.openjudge.cn/2024fallroutine/23564">http://cs101.openjudge.cn/2024fallroutine/23564</a>
	460A. Vasya and Socks	brute force, implementation, math	900	<a href="https://codeforces.com/contest/460/problem/A">https://codeforces.com/contest/460/problem/A</a>
0909	1374B. Multiply by 2, divide by 6	math	900	<a href="https://codeforces.com/problemset/problem/1374/B">https://codeforces.com/problemset/problem/1374/B</a>
	01003: Hangover	-	-	<a href="http://cs101.openjudge.cn/2024fallroutine/01003/">http://cs101.openjudge.cn/2024fallroutine/01003/</a>
0908	18223: 24点	-	-	<a href="http://cs101.openjudge.cn/2024fallroutine/18223/">http://cs101.openjudge.cn/2024fallroutine/18223/</a>
	21532: 数学密码	-	-	<a href="http://cs101.openjudge.cn/2024fallroutine/21532/">http://cs101.openjudge.cn/2024fallroutine/21532/</a>
0907	19949: 提取实体	-	-	<a href="http://cs101.openjudge.cn/2024fallroutine/19949/">http://cs101.openjudge.cn/2024fallroutine/19949/</a>
	02808: 校门外的树	-	-	<a href="http://cs101.openjudge.cn/2024fallroutine/02808/">http://cs101.openjudge.cn/2024fallroutine/02808/</a>
0906	12566: 编码字符串	-	-	<a href="http://cs101.openjudge.cn/2024fallroutine/12556/">http://cs101.openjudge.cn/2024fallroutine/12556/</a>
	06374: 文字排版	-	-	<a href="http://cs101.openjudge.cn/2024fallroutine/06374/">http://cs101.openjudge.cn/2024fallroutine/06374/</a>
0905	1833B. Restore the Weather	greedy, sortings	900	<a href="https://codeforces.com/problemset/problem/1833/B">https://codeforces.com/problemset/problem/1833/B</a>
	1879B. Chips on the Board	constructive algorithms, greedy	900	<a href="https://codeforces.com/contest/1879/problem/B">https://codeforces.com/contest/1879/problem/B</a>
0904	96A. Football	implementation, strings	900	<a href="https://codeforces.com/problemset/problem/96/A">https://codeforces.com/problemset/problem/96/A</a>
	160A. Twins	greedy, sortings	900	<a href="https://codeforces.com/problemset/problem/160/A">https://codeforces.com/problemset/problem/160/A</a>
0903	1475A. Odd Divisor	math, number theory	900	<a href="https://codeforces.com/problemset/problem/1475/A">https://codeforces.com/problemset/problem/1475/A</a>
	34B. Sale	greedy, sortings	900	<a href="https://codeforces.com/problemset/problem/34/B">https://codeforces.com/problemset/problem/34/B</a>
0902	25A. IQ test	brute force	1300	<a href="https://codeforces.com/problemset/problem/25/A">https://codeforces.com/problemset/problem/25/A</a>
	281A. Word Capitalization	implementation, strings	800	<a href="http://codeforces.com/problemset/problem/281/A">http://codeforces.com/problemset/problem/281/A</a>
0901	474A. ...	implementation	900	<a href="http://codeforces.com/problemset/problem/474/A">http://codeforces.com/problemset/problem/474/A</a>

## 21532: 数学密码

Brute force, <http://cs101.openjudge.cn/practice/21532>

小明和同学一起去玩密室逃脱，这个密室的老板特别喜欢数学，所以他设置了一个密码门。在搜集了房间里的有价值信息后，小明总结出的信息为：三个互不相同的正整数和为231，那么这三个互不相同正整数的最大公因数就是密码。小明很快的算出了密码。

回到学校后，小明希望写一个程序，对于给定的三个互不相同的正整数的和，快速求出这三个正整数的最大公因数。

描述部分对应英文：

Xiao Ming went to play escape room with his classmates. The owner of this room is especially fond of mathematics, so he set up a password door. After collecting valuable information from the room, Xiao Ming concluded the information as follows: three mutually dissimilar positive integers and 231, then the greatest common factor of these three mutually dissimilar positive integers is the password. Xiao Ming quickly worked out the password.

After returning to school, Xiao Ming wishes to write a program to quickly find the greatest common factor of the three mutually dissimilar positive integers for a given sum of these three positive integers.

## 输入

输入为三个互不相同的正整数的和 ( $6 \leq N \leq 109$ ).

## 输出

输出三个数的最大公因数.

## 样例输入

```
1 sample1 in:  
2 231  
3  
4 sample1 out:  
5 33
```

## 样例输出

```
1 sample2 in:  
2 1234  
3  
4 sample2 out:  
5 2  
6  
7 sample3 in:  
8 88  
9  
10 sample3 out:  
11 11
```

来源: cs101 2020 Final Exam

出这个题目的思路，来自学而思小学高年级数学的一次板书。

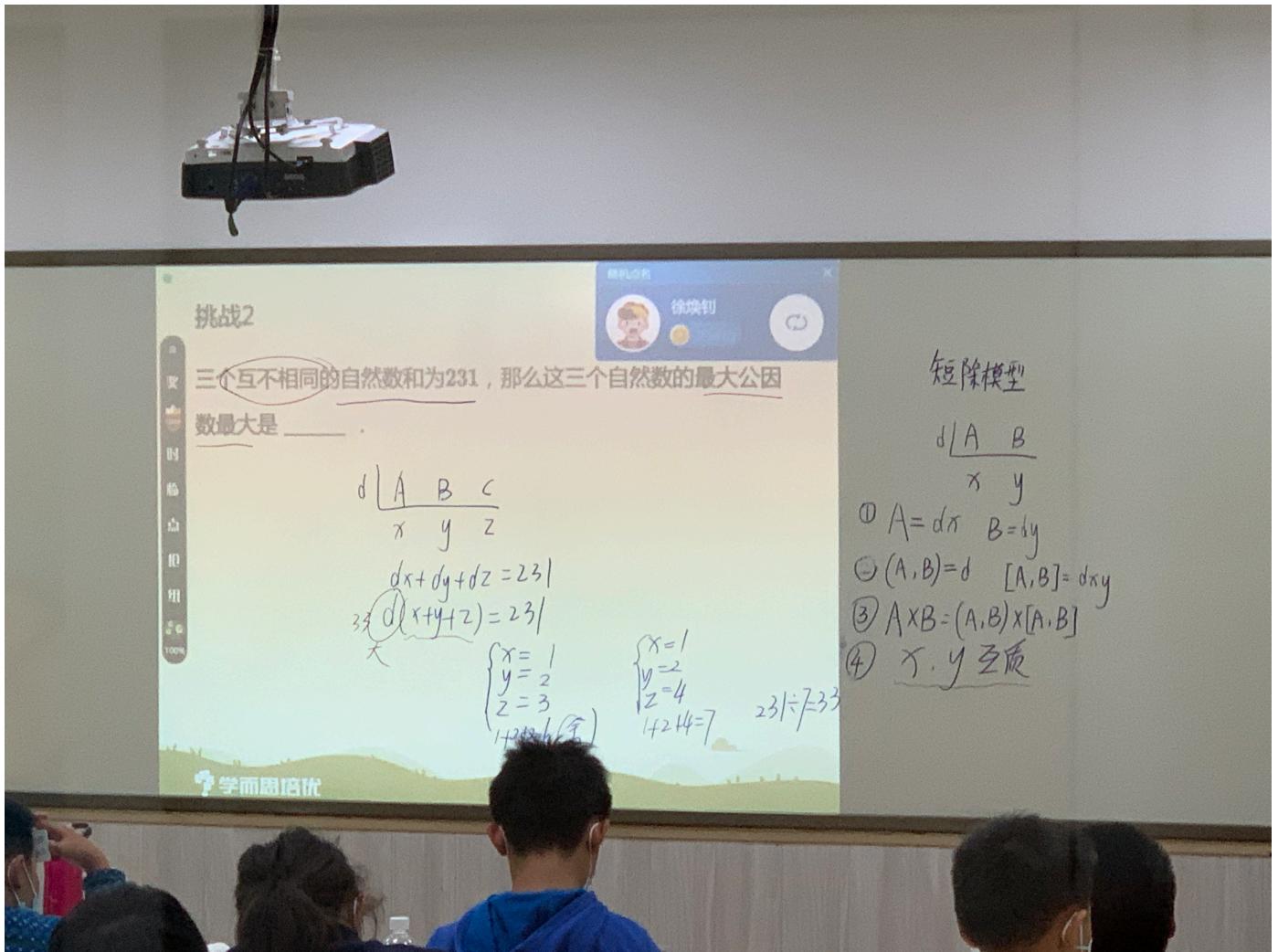


图 一个小学高年级数学题目

对于大于等于6的整数，可以表示为三个互不相同的正整数的和。可以从1和2开始。由于 $S-3$ 必定大于2，所以我们确保这三个数互不相同。

## 2024fall-cs101: Algo DS (415)



俞天麒

你有没有发现其实这题只要算他某个条件下最大的因数就行了



俞天麒

有时候不用题目说什么做什么

Yesterday 18:18



24 数院 谢昊宸

正确的



24 数院 谢昊宸

找一个很大的因数即可



24 物-唐晨宇

只要是除法得到的商比6大



24 物-唐晨宇

就行



24 物-唐晨宇

整除

2024/9/9

```
1 n=int(input())
2 i = 1 + 2 + 3
3 while n%i != 0:
4     i += 1
5 else:
6     print(n // i)
```

数学思维 vs 计算机思维，即抽象、简化 vs 分解、执行

这种方法体现了深刻的数学洞察力，可能不是短时间内就能想到的。在解决这类问题时，用计算机思维的枚举策略同样能够获得正确的答案 (AC)。如果超时了，可以考虑改变遍历方向。OJ评测系统中题目的测试数据，通常比较弱，考试时候给出的测试数据一般不超过20组。

```
1 t = int(input())
```

```

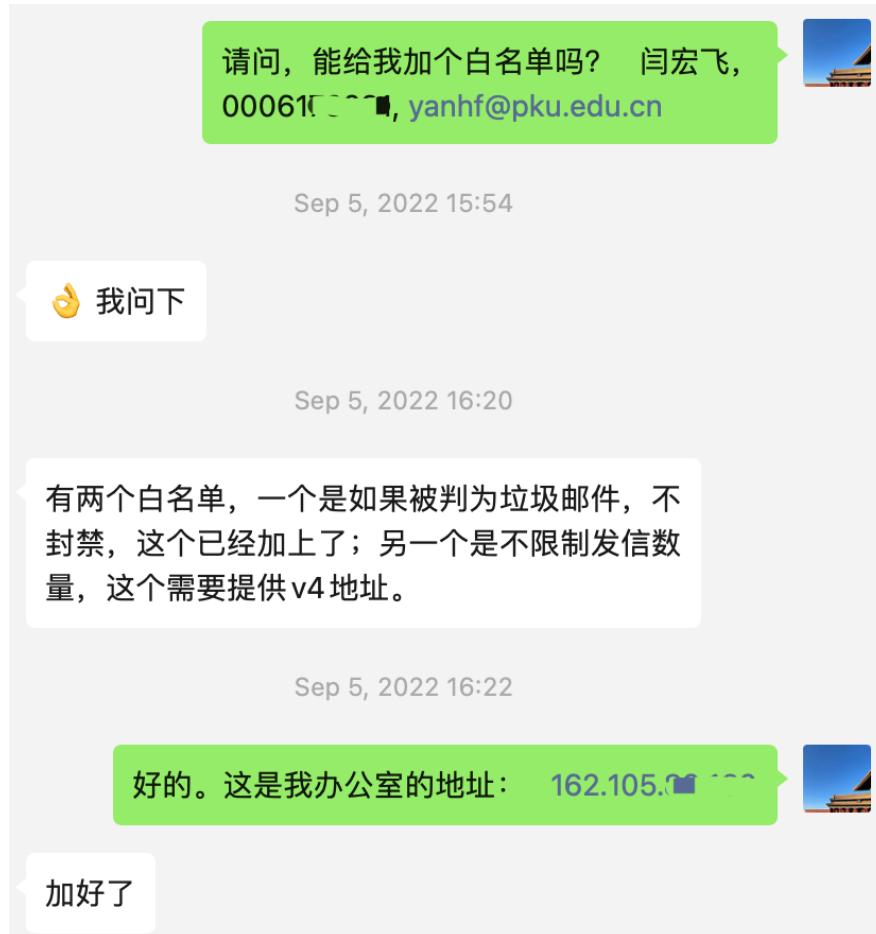
2
3 s = 1 + 2 +3
4
5 while True:
6     while t%s !=0:
7         s += 1
8         ...
9         if s > t**0.5:
10            t = s
11            break
12        ...
13
14 found = False
15 for x in range(1, s):
16     for y in range(x+1, s):
17         if y==x:
18             continue
19         for z in range(y+1, s):
20             if s - x - y != z:
21                 continue
22             if z==y or z==x:
23                 continue
24
25             #print(f'{x},{y},{z}')
26             found = True
27             break
28
29         if found:
30             break
31     if found:
32         break
33
34 if found:
35     break
36
37 print(t//s)

```

# 四、其他程序

## 4.1 发邮件给选课同学

<https://github.com/GMyhf/2024fall-cs101/blob/main/code/sendEmail.py>



## 4.2 选课学生院系分布

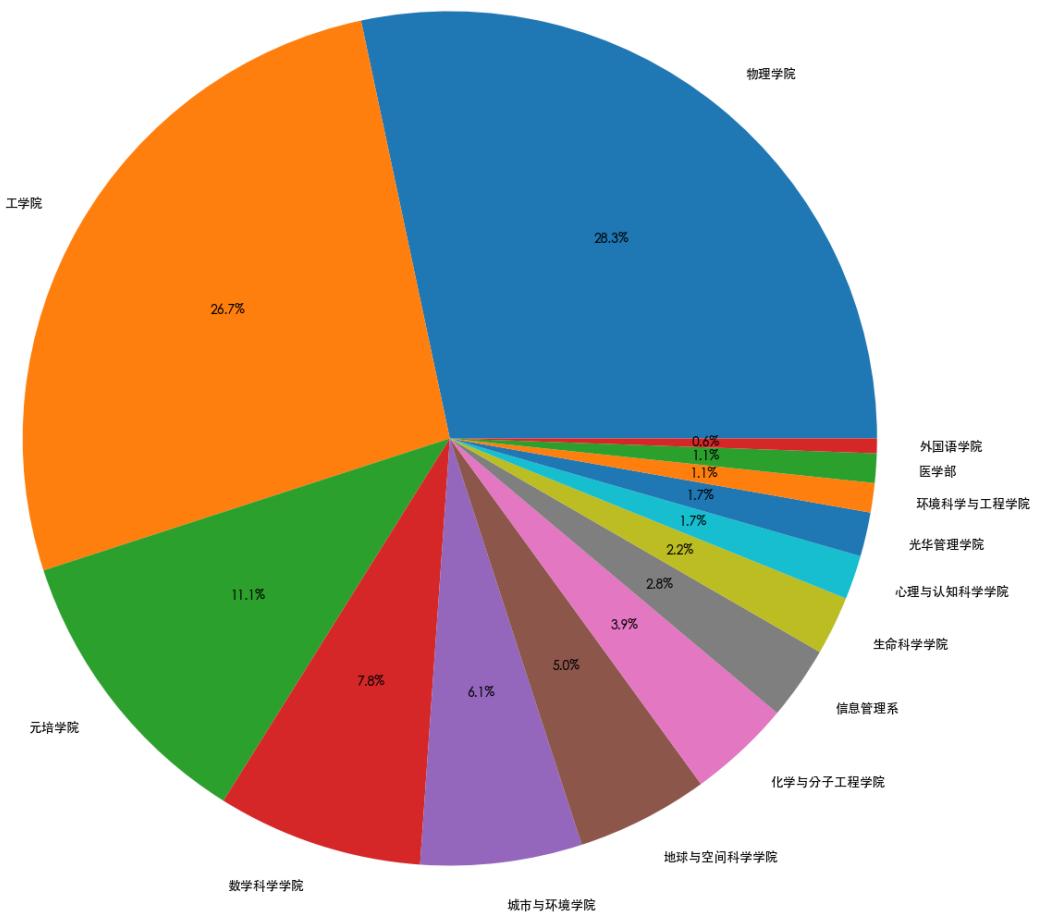
```
1 import pandas as pd
2
3 df = pd.read_excel('studentListInCourse-20240907.xls')
4 df_schools = df['所属院系'].value_counts().reset_index()
5 df_schools.columns = ['所属院系', 'Count']
6 #print(df_schools)
7
8 import itertools
9 #print(*df_schools.loc[:, ['所属院系']].values)
10 #print(*df_schools.loc[:, ['Count']].values)
11 labels = list(itertools.chain(*df_schools.loc[:, ['所属院系']].values))
12 sizes = list(itertools.chain(*df_schools.loc[:, ['Count']].values))
13
14 import matplotlib.pyplot as plt
```

```

15 plt.rcParams['font.family'] = ['Heiti TC']
16
17 fig, ax = plt.subplots(figsize=(15, 15))
18 ax.pie(sizes, labels=labels, autopct='%1.1f%%')
19
20 plt.title(label="2024/9/7 选课院系分布",
21             fontsize=20,
22             loc="left",
23             color="green")
24 plt.show()

```

2024/9/7 选课院系分布



## 4.3 洛谷运势

在洛谷打卡, <https://www.luogu.com.cn>, 有个运势, 如图。



程序实现如下。

<https://github.com/GMyhf/2024fall-cs101/blob/main/code/fortune.py>

```
1 # 参考 https://blog.csdn.net/pythonwyw123321/article/details/136659699
2 import random
3
4 # 定义随机函数
5 def get_random_int(a, b):
6     return random.randint(a, b)
7
8 # 运势权重列表
9 weights = [2, 4, 15, 15, 16, 16, 25, 7, 5]
10 # 运势名称
11 fortunes = [
12     "宇宙超级凶", "大凶", "中平", "小平",
13     "小凶", "中吉", "小吉", "超级吉", "中凶"
14 ]
15
16 # 宜做的事情列表
17 yi_list = [
18     ["宜:诸事不宜"] * 4,
19     ["宜:装弱", "宜:窝在家里", "宜:刷题", "宜:吃饭"],
20     ["宜:刷题", "宜:开电脑", "宜:写作业", "宜:睡觉"],
21     ["宜:发朋友圈", "宜:出去玩", "宜:打游戏", "宜:吃饭"],
22     ["宜:学习", "宜:研究Ruby", "宜:研究c#", "宜:玩游戏"],
23     ["宜:膜拜大神", "宜:扶老奶奶过马路", "宜:玩网游", "宜:喝可乐"],
24     ["宜:吃东西", "宜:打sdvx", "宜:打开洛谷", "宜:出行"],
25     ["宜:写程序", "宜:刷题", "宜:偷塔", "宜:上CSDN"],
26     ["宜:扶老奶奶过马路", "宜:上课", "宜:写作业", "宜:写程序"]
27 ]
28
29 # 宜做的事情解释列表
```

```

30 yi_shi_list = [
31     [ "" ] * 4,
32     [ "谦虚最好了", "不出门没有危险", "直接AC", "吃的饱饱的再学习"],
33     [ "一次AC", "发现电脑死机了", "全对", "睡足了再学习"],
34     [ "点赞量破百", "真开心", "十连胜", "吃饱了"],
35     [ "都会", "有了新发现", "发现新大陆", "直接胜利"],
36     [ "接受神之沐浴", "增加RP", "犹如神助", "真好喝"],
37     [ "吃饱了", "今天状态好", "发现AC的题变多了", "路途顺畅"],
38     [ "不会报错", "直接TLE", "胜利", "发现粉丝涨了200个"],
39     [ "增加RP", "听懂了", "都会", "没有Bug"]
40 ]
41
42 # 忌做的事情列表
43 ji_list = [
44     [ "忌:诸事不宜" ] * 4,
45     [ "忌:打sdvx", "忌:出行", "忌:玩手机", "忌:吃方便面"],
46     [ "忌:关电脑", "忌:开挂", "忌:纳财", "忌:考试"],
47     [ "忌:膜拜大神", "忌:评论", "忌:研究Java", "忌:吃方便面"],
48     [ "忌:发朋友圈", "忌:打开洛谷", "忌:研究C++", "忌:出行"],
49     [ "忌:探险", "忌:发视频", "忌:发博客", "忌:给别人点赞"],
50     [ "忌:写程序", "忌:使用Unity打包exe", "忌:装弱", "忌:打开CSDN"],
51     [ "忌:点开wx", "忌:刷题", "忌:打吃鸡", "忌:和别人分享你的程序"],
52     [ "忌:纳财", "忌:写程序超过500行", "忌:断网", "忌:检测Bug"]
53 ]
54
55 # 忌做的事情解释列表
56 ji_shi_list = [
57     [ "" ] * 4,
58     [ "今天状态不好", "路途也许坎坷", "好家伙, 直接死机", "没有调味料"],
59     [ "死机了", "被制裁", "你没有财运", "没及格"],
60     [ "被人嘲笑", "被喷", "心态崩溃", "只有一包调味料"],
61     [ "被人当成买面膜的", "大凶", "五行代码198个报错", "路途坎坷"],
62     [ "你失踪了", "被人喷", "阅读量1", "被人嘲笑"],
63     [ "报错19999+", "电脑卡死, 发现刚才做的demo全没了", "被人看穿", "被人陷害"],
64     [ "被人陷害", "WA", "被队友炸死", "别人发现了Bug"],
65     [ "没有财运", "99+报错", "连不上了", "503个Bug"]
66 ]
67
68 # 计算权重总和
69 total_weight = sum(weights)
70 # 生成随机数
71 random_value = random.randint(0, total_weight)
72 fortune_index = 0
73
74 # 根据权重选择运势
75 for weight in weights:
76     if random_value <= weight:
77         break
78     random_value -= weight
79     fortune_index += 1
80
81 # 输出运势结果

```

```
82 print("\t\t你的运势是:")
83 print("\t\t" + fortunes[fortune_index])
84 for idx, name in enumerate(fortunes):
85     if name == fortunes[fortune_index]:
86         print("\t" + yi_list[idx][get_random_int(0, 3)], end="")
87         print("\t" + ji_list[idx][get_random_int(0, 3)])
88         print("\t" + yi_shi_list[idx][get_random_int(0, 3)], end="")
89         print("\t" + ji_shi_list[idx][get_random_int(0, 3)])
90         break
```

# 参考

---

2023/11/5，我用这个方法，<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/reliablyang/6457392>

根据各人情况，选择一个，加入hosts文件

```
#185.199.108.133 raw.githubusercontent.com  
185.199.109.133 raw.githubusercontent.com  
185.199.110.133 raw.githubusercontent.com  
185.199.111.133 raw.githubusercontent.com
```