

Assignment #1: 虚拟机, Shell & 大语言模型

Updated 2309 GMT+8 Feb 20, 2025

2025 spring, Compiled by 叶靖、信息管理学院

作业的各项评分细则及对应的得分

标准	等级	得分
按时提交	完全按时提交：1分 提交有请假说明：0.5分 未提交：0分	1分
源码、耗时（可选）、解题思路（可选）	提交了4个或更多题目且包含所有必要信息：1分 提交了2个或以上题目但不足4个：0.5分 少于2个：0分	1分
AC代码截图	提交了4个或更多题目且包含所有必要信息：1分 提交了2个或以上题目但不足4个：0.5分 少于：0分	1分
清晰头像、PDF文件、MD/DOC附件	包含清晰的Canvas头像、PDF文件以及MD或DOC格式的附件：1分 缺少上述三项中的任意一项：0.5分 缺失两项或以上：0分	1分
学习总结和个人收获	提交了学习总结和个人收获：1分 未提交学习总结或内容不详：0分	1分
总得分： 5	总分满分：5分	

说明：

1. 解题与记录：

- 对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge，Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora <https://typoraio.cn> 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。

2. 课程平台与提交安排：

- 我们的课程网站位于Canvas平台（<https://pku.instructure.com>）。该平台将在第2周选课结束后正式启用。在平台启用前，请先完成作业并将作业妥善保存。待Canvas平台激

活后，再上传你的作业。

- 提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。

3. 延迟提交：

- 如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

27653: Fraction类

<http://cs101.openjudge.cn/practice/27653/>

思路：

此题算是基础题目，只需要知道Fraction包，即可直接进行分数的简单运算。

代码：

```
from fractions import Fraction f1_0, f1_1, f2_0, f2_1 = list(map(int, input().split())) ans = Fraction(f1_0, f1_1) + Fraction(f2_0, f2_1) # print(ans)
```

代码运行截图（至少包含有"Accepted"）



1760.袋子里最少数目的球

<https://leetcode.cn/problems/minimum-limit-of-balls-in-a-bag/>

思路：

题目要求我们将一个袋子的球分成两个袋子时，最后分配结果的最大球数尽可能地小。通过二分查找的概念优化过程，检查能否通过maxOperations将最大球数限制为我们所求结果。此题难度不大，类似的题目在计概课程中没少出现，但由于很少使用Leetcode不大适应其环境，需要将程序嵌套其中，所以耗了点时间在这过程中。

代码：

```
class Solution: def minimumSize(self, nums: List[int], maxOperations: int) -> int: def min_operations(nums, maxOperations): left, right = 1, int(max(nums)) while left < right: mid = (left + right) // 2 operations_needed = 0 for num in nums: if num > mid: operations_needed +=
```

```
(num - 1) // mid if operations_needed <= maxOperations: right = mid else: left = mid + 1 return
left return min_operations(nums, maxOperations)
```

代码运行截图 （至少包含有"Accepted"）



04135: 月度开销

<http://cs101.openjudge.cn/practice/04135>

思路：

和上一题差不多，都是想将最大的分组结果最小化，使用的是二分查找加上贪心检查，尽可能少分组的同时最小化分组结果。题目难度算中等，但是需要不断改良来减少耗时。

代码：

```
def checking(expenses, N, M, max_limit): count = 1 current_sum = 0
```

```
    for expense in expenses:
        if current_sum + expense > max_limit:
            count += 1
            current_sum = expense
            if count > M:
                return False
        else:
            current_sum += expense
    return True
```

```
def min_expenses(N, M, expenses): left = max(expenses) right = sum(expenses) while left < right:
    max_limit = (left + right) // 2 if checking(expenses, N, M, max_limit): right = max_limit else: left =
    max_limit + 1 return left
```

```
N, M = map(int, input().split()) expenses = [] for _ in range(N): expenses.append(int(input()))
print(min_expenses(N, M, expenses))
```

代码运行截图 （至少包含有"Accepted"）



27300: 模型整理

<http://cs101.openjudge.cn/practice/27300/>

思路：

此题的难度我认为不会太高，唯一的卡点就是在单位换算（M及B）的地方，因为题目需要根据大小顺序排列数据。

代码：

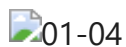
```
def splitting(model_str): type, size_str = model_str.rsplit('-', 1) if size_str.endswith('M'): model_value = float(size_str[:-1]) else: model_value = float(size_str[:-1]) * 1000 return type, model_value, size_str
```

```
def sorting_model(n, models): model_dict = {} for model in models: type, value, size = splitting(model) if type not in model_dict: model_dict[type] = [] model_dict[type].append((value, size))
```

```
for type in sorted(model_dict.keys()):
    sizes = model_dict[type]
    sorted_sizes = sorted(sizes, key=lambda x: x[0])
    formatted_sizes = [size_str for _, size_str in sorted_sizes]
    print(f"{type}: {' '.join(formatted_sizes)}")
```

```
n = int(input()) models = [input().strip() for _ in range(n)] sorting_model(n, models)
```

代码运行截图（至少包含有"Accepted"）



Q5. 大语言模型（LLM）部署与测试

本任务旨在本地环境或通过云虚拟机（如 <https://clab.pku.edu.cn/> 提供的资源）部署大语言模型（LLM）并进行测试。用户界面方面，可以选择使用图形界面工具如 <https://lmstudio.ai> 或命令行界面如 <https://www.ollama.com> 来完成部署工作。

测试内容包括选择若干编程题目，确保这些题目能够在所部署的LLM上得到正确解答，并通过所有相关的测试用例（即状态为Accepted）。选题应来源于在线判题平台，例如 OpenJudge、Codeforces、LeetCode 或洛谷等，同时需注意避免与已找到的AI接受题目重复。已有的AI接受题目列表可参考以下链接：https://github.com/GMyhf/2025spring-cs201/blob/main/AI_accepted_locally.md

请提供你的最新进展情况，包括任何关键步骤的截图以及遇到的问题和解决方案。这将有助于全面了解项目的推进状态，并为进一步的工作提供参考。

Q6. 阅读《Build a Large Language Model (From Scratch)》第一章

作者：Sebastian Raschka

请整理你的学习笔记。这应该包括但不限于对第一章核心概念的理解、重要术语的解释、你认为特别有趣或具有挑战性的内容，以及任何你可能有的疑问或反思。通过这种方式，不仅能巩固你自己的学习成果，也能帮助他人更好地理解这一部分内容。

2. 学习总结和个人收获

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

此作业的重点集中于二分查找及指针类型的题目。我认为这份作业中需要敲代码的题目都不是太难，应该算是让我们先找回手感，遇到的卡点一般都还是能在较短的时间内处理。可能是刚开学，作业量还不是特别重，目前的每日练习进度尚可，在空闲时都有上去尝试。最后还有一点可能需要麻烦老师，如果我的作业的格式出了什么问题，劳烦老师能够及时提醒我，让我能在下次的作业中进行改善。