

# assignment1

## Assignment #1: 虚拟机, Shell & 大语言模型

2025 fall, Complied by 杨知进 物理学院

### 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：

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。

3. **提交安排：**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本人头像，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。
4. **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

### E27653: Fraction类

<http://cs101.openjudge.cn/pctbook/E27653/>

请练习用OOP方式实现。

思路：

代码：

```
import math

class Fraction:
    def __init__(self, up, down):
        if down == 0:
            raise ValueError("Could not divided by 0")
        elif down < 0:
            down = -1*down
            up = -1*up
        factor = math.gcd(up, down)
        self.up = up//factor
        self.down = down//factor
    def __add__(self, other):
        new_up = self.up*other.down+self.down*other.up
        new_down = self.down*other.down
        factor = math.gcd(new_up, new_down)
        return Fraction(new_up//factor, new_down//factor)
    def __str__(self):
        return f'{self.up}/{self.down}'
a, b, c, d = map(int, input().split())
print(Fraction(a, b)+Fraction(c, d))
```

代码运行截图 (至少包含有"Accepted")



#49996104提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

import math

class Fraction:
    def __init__(self, up, down):
        if down == 0:
            raise ValueError("Could not divided by 0")
        elif down < 0:
            down = -1*down
            up = -1*up
        factor = math.gcd(up, down)
        self.up = up//factor
        self.down = down//factor
    def __add__(self, other):
        new_up = self.up*other.down+self.down*other.up
        new_down = self.down*other.down
        factor = math.gcd(new_up, new_down)
        return Fraction(new_up//factor, new_down//factor)
    def __str__(self):
        return f'{self.up}/{self.down}'
a, b, c, d = map(int, input().split())
print(Fraction(a, b)+Fraction(c, d))

```

## 基本信息

#: 49996104  
 题目: E27653  
 提交人: 2400011575  
 内存: 3640kB  
 时间: 23ms  
 语言: Python3  
 提交时间: 2025-09-14 23:00:23

用时: 4min左右

## M1760.袋子里最少数目的球

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

思路:

二分查找，初始上限是原袋子中最多的球数，下限是1。对每个初始的袋子，要拆分到mid数目以下需要操作次数为( $\text{num}-1$ )//mid。需要的操作次数大于maxOperations表明mid过小，否则表明mid过大。

代码:

```

class Solution:
    def minimumSize(self, nums: List[int], maxOperations: int) -> int:
        low, high = 1, max(nums)
        while low <= high:
            mid = (low + high) // 2
            count = 0
            for num in nums:
                count += (num - 1) // mid
                if count > maxOperations:

```

```

        low = mid + 1
        break
    else:
        high = mid - 1
return low

```

代码运行截图 (至少包含有"Accepted")

```

</> 代码
Python3 智能模式

1 class Solution:
2     def minimumSize(self, nums: List[int], maxOperations: int) -> int:
3         low, high = 1, max(nums)
4         while low <= high:
5             mid = (low + high) // 2
6             count = 0
7             for num in nums:
8                 count += (num - 1) // mid
9                 if count > maxOperations:
10                    low = mid + 1
11                else:
12                    high = mid - 1
13         return low

```

已存储

题目描述 | 通过 × | 题解 | 提交记录

← 全部提交记录

通过 61 / 61 个通过的测试用例

官方题解 写题解

用时：10min左右

## M04135: 月度开销

binary search, <http://cs101.openjudge.cn/pctbook/M04135/>

思路：

二分查找，初始上限是所有开销之和，下限是单个开销的最大值。分月份的次数大于m表明该开销过小，否则表明该开销过大。

代码：

```

n, m = map(int, input().split())
nums = [int(input()) for _ in range(n)]

def check(x):

```

```
count = 1
current_sum = 0
for num in nums:
    if current_sum + num <= x:
        current_sum += num
    else:
        count += 1
        current_sum = num
    if count > m:
        return False
return True
low = max(nums)
high = sum(nums)

while low <= high:
    mid = (low + high) // 2
    if check(mid):
        high = mid - 1
    else:
        low = mid + 1
print(low)
```

代码运行截图 (至少包含有"Accepted")



## #49996197提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

n, m = map(int, input().split())
nums = [int(input()) for _ in range(n)]

def check(x):
    count = 1
    current_sum = 0
    for num in nums:
        if current_sum + num <= x:
            current_sum += num
        else:
            count += 1
            current_sum = num
        if count > m:
            return False
    return True

low = max(nums)
high = sum(nums)

while low <= high:
    mid = (low + high) // 2
    if check(mid):
        high = mid - 1
    else:
        low = mid + 1
print(low)

```

## 基本信息

#: 49996197  
 题目: M04135  
 提交人: 2400011575  
 内存: 7500kB  
 时间: 381ms  
 语言: Python3  
 提交时间: 2025-09-14 23:15:30

用时: 15min左右

## M27300: 模型整理

sortings, AI, <http://cs101.openjudge.cn/pctbook/M27300/>

思路:

使用字典存储model名称与值 (保留原始字符和转化单位后的数值), 先按名称排序, 再按数值排序, 输出。

代码:

```

n = int(input().strip())
models = {}
for _ in range(n):
    s = input().strip()
    name, size = s.split("-")
    value = float(size[:-1]) * (1e3 if size[-1] == 'B' else 1)
    models.setdefault(name, []).append((size, value))

```

```

for name in sorted(models.keys()):
    sorted_size = sorted(models[name], key = lambda x: x[-1])
    size = ", ".join(size for size,value in sorted_size)
    print(name + ": " + size)

```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

2400011575 信箱 账号

CS101 / 计算思维算法实践

题目 排名 状态 提问

#49993968 提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 49993968  
题目: M27300  
提交人: wanqian  
内存: 3620kB  
时间: 21ms  
语言: Python3  
提交时间: 2025-09-14 20:03:29

源代码

```

n = int(input().strip())
models = {}
for _ in range(n):
    s = input().strip()
    name, size = s.split("-")
    value = float(size[:-1]) * (1e3 if size[-1] == 'B' else 1)
    models.setdefault(name, []).append((size, value))

for name in sorted(models.keys()):
    sorted_size = sorted(models[name], key = lambda x: x[-1])
    size = ", ".join(size for size,value in sorted_size)
    print(name + ": " + size)

```

用时: 6min左右

## Q5. 熟悉云虚拟机Linux环境与大语言模型 (LLM) 本地部署

本项目包括两个任务:

- 1) 通过云虚拟机 (如 <https://clab.pku.edu.cn/> 提供的资源) 熟悉Linux系统操作环境。
- 2) 完成大语言模型 (LLM) 的本地部署与功能测试。

LLM 部署可选择使用图形化工具 (如 LM Studio, <https://lmstudio.ai>) 以简化配置流程, 提升部署效率。部署完成后, 需对模型进行实际能力测试。

测试内容包括: 从主流在线编程评测平台 (如 OpenJudge、Codeforces、LeetCode 或洛谷等) 选取若干编程题目, 提交由本地部署的 LLM 生成的代码解决方案, 并确保其能够通过全部测试用例, 获得“Accepted”状态。选题时应避免与已知可被 AI 正确解答的题目重复。当前已确认可通过的 AI 解题列表可参考以下 GitHub 仓库:

[https://github.com/GMyhf/2025spring-cs201/blob/main/AI\\_accepted\\_locally.md](https://github.com/GMyhf/2025spring-cs201/blob/main/AI_accepted_locally.md)

请提供你的项目进展, 内容应该包括: 关键操作步骤的截图以及遇到的技术问题及相应的解决方法。这将有助于全面掌握项目推进情况, 并为后续优化与扩展提供依据。

## **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、关键术语的解析、学习过程中的思考与启发，以及尚存的疑问与反思。通过系统梳理，不仅有助于巩固自身理解，也希望为其他学习者提供有价值的参考。

## **2. 学习总结和个人收获**

距离计概已经过去一个学期没怎么写过代码，重新熟悉的同时也学习了很多之前没有了解过的算法（比如反复出现的二分查找）。找到一个算法学习网站[Hello 算法](#)，感觉讲的很清晰，弥补一下上学期模模糊糊的动态规划。人工智能这一块暂时可能抽不出太多时间。