

Assignment #1: 虚拟机，Shell & 大语言模型

Updated 2309 GMT+8 Feb 20, 2025

2025 spring, Compiled by <mark>胡新璞, 工学院/mark>

1. 题目

27653: Fraction 类

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

思路：简单的数学运算

代码：

```
from math import gcd
a,b,c,d = map(int,input().split())
fenmu = int(b * d / gcd(b,d))
fenzi = int(a * (d / gcd(b,d)) + c * (b / gcd(b,d)))
print(str(fenzi) + "/" + str(fenmu))
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
from math import gcd
a,b,c,d = map(int,input().split())
fenmu = int(b * d / gcd(b,d))
fenzi = int(a * (d / gcd(b,d)) + c * (b / gcd(b,d)))
print(str(fenzi) + "/" + str(fenmu))
```

基本信息

#: 48434965

题目: 27653

提交人: 2400011037

内存: 3612kB

时间: 28ms

语言: Python3

提交时间: 2025-03-04 16:41:56

1760.袋子里最少数目的球

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

思路：二分查找，和 maxO 进行比较。

代码：

class Solution(object):

```
    def minimumSize(self, nums, maxOperations):
```

```
        l, r = 1, max(nums) + 1
```

```
        ans = 0
```

```
        while l < r:
```

```
            operations = 0
```

```
            n = (l + r) // 2
```

```
            for i in range(len(nums)):
```

```
                operations += (nums[i] - 1) // n
```

```
            if operations <= maxOperations:                # "="
```

```
                r = n
```

```
                ans = n
```

```
            else:
```

```
                l = n + 1
```

```
        return ans
```

代码运行截图（至少包含有"Accepted"）

The screenshot displays the LeetCode interface for problem 1760. On the left, it shows the problem title, a success message '通过 60 / 60 个通过的测试用例', and submission details. A banner for '面向在校学生的专享特惠' is visible. Below, the '执行用时分布' (Execution Time Distribution) section shows a runtime of 639 ms and a击败率 of 27.10%. On the right, the code is shown in a text editor, and the '测试结果' (Test Results) section at the bottom indicates '已通过' (Accepted) for all test cases.

04135: 月度开销

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

思路：参照题解。一寒假没学现在对我来说有点太难了。

代码：

```
n,m = map(int, input().split())
expenditure = []
for _ in range(n):
    expenditure.append(int(input()))
```

```
def check(x):
    num, s = 1, 0
    for i in range(n):
        if s + expenditure[i] > x:
            s = expenditure[i]
            num += 1
        else:
            s += expenditure[i]
    return [False, True][num > m]
```

```
lo = max(expenditure)
hi = sum(expenditure) + 1
ans = 1
while lo < hi:
    mid = (lo + hi) // 2
    if check(mid):
        lo = mid + 1
    else:
        ans = mid
        hi = mid

print(ans)
```

代码运行截图 （至少包含有"Accepted"）

状态: **Accepted**

源代码

```
n,m = map(int, input().split())
expenditure = []
for _ in range(n):
    expenditure.append(int(input()))

def check(x):
    num, s = 1, 0
    for i in range(n):
```

基本信息

#: 48520358
题目: 04135
提交人: 2400011037
内存: 7996kB
时间: 541ms
语言: Python3
提交时间: 2025-03-11 15:24:26

27300: 模型整理

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

思路：函数 $s(x)$ 用来进行对参数大小的排序，llm 字典用来确定模型数量并按模型排序

代码：

```
def s(x):
    x1, x2 = [], []
    for i in range(len(x)):
        x1.append((x[i][:-1])) if x[i][-1] == "M" else x2.append((x[i][:-1]))
    x1.sort(key = lambda x : float(x))
    x2.sort(key = lambda x : float(x))
    for i in range(len(x1)):
        x1[i] = str(x1[i]) + "M"
    for j in range(len(x2)):
        x2[j] = str(x2[j]) + "B"
    return x1 + x2
```

```
n = int(input())
llm = {}
for _ in range(n):
    a,b = map(str,input().split("-"))
    if a in llm:
        llm[a] = s(llm[a] + [b])
    else:
        llm[a] = [b]
llm = sorted(llm.items())
for _ in range(len(llm)):
    print(llm[_][0] + ": " + ", ".join(map(str, llm[_][1])))
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
def s(x):
    x1, x2 = [], []
    for i in range(len(x)):
        x1.append((x[i][:-1])) if x[i][-1] == "M" else x2.append((x[i][:-1]))
    x1.sort(key = lambda x : float(x))
    x2.sort(key = lambda x : float(x))
    for i in range(len(x1)):
        x1[i] = str(x1[i]) + "M"
    for j in range(len(x2)):
        x2[j] = str(x2[j]) + "B"
    return x1 + x2

n = int(input())
llm = {}
for _ in range(n):
    a,b = map(str,input().split("-"))
    if a in llm:
        llm[a] = s(llm[a] + [b])
    else:
        llm[a] = [b]
llm = sorted(llm.items())
for _ in range(len(llm)):
    print(llm[_][0] + ": " + ", ".join(map(str, llm[_][1])))
```

基本信息

#: 48520303
题目: 27300
提交人: 2400011037
内存: 3696kB
时间: 33ms
语言: Python3
提交时间: 2025-03-11 15:20:20

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。

作业题有点难，二分查找上学期就没学太明白，这学期写下来依然需要反复 debug，或者对着题解调整思路。大模型还没有来得及学，虚拟机的相关知识尝试了一点，努力追赶中。