

Assignment #7: Nov Mock Exam 立冬

Updated 1646 GMT+8 Nov 7, 2024

2024 fall, Compiled by <mark>同学的姓名、院系</mark>

说明:

1) 月考: AC6<mark> (请改为同学的通过数) </mark>。考试题目都在“题库 (包括计概、数算题目)”里面, 按照数字题号能找到, 可以重新提交。作业中提交自己最满意版本的代码和截图。

2) 请把每个题目解题思路 (可选), 源码 Python, 或者 C++ (已经在 Codeforces/Openjudge 上 AC), 截图 (包含 Accepted), 填写到下面作业模版中 (推荐使用 typora <https://typoraio.cn>, 或者用 word)。AC 或者没有 AC, 都请标上每个题目大致花费时间。

3) 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”。Canvas 需要有同学清晰头像、提交文件有 pdf、“作业评论”区有上传的 md 或者 doc 附件。

4) 如果不能在截止前提交作业, 请写明原因。

1. 题目

E07618: 病人排队

sorttings, <http://cs101.openjudge.cn/practice/07618/>
(30min)

思路: 先输入病人信息, 将病人 ID、年龄和输入顺序组成一个元组, 添加到 `patients` 列表中, 然后对病人信息进行排序, 最后输出即可。

代码:

```
n = int(input())
patients = []
for i in range(n):
    patient_info = input().split()
    patient_id = patient_info[0]
    age = int(patient_info[1])
```

```
patients.append((patient_id, age, i))

patients.sort(key=lambda x: (-x[1] >= 60, -x[1] if
x[1] >= 60 else x[2]))

for patient in patients:
    print(patient[0])
```

代码运行截图 [<mark>](#)（至少包含有"Accepted"）[</mark>](#)

状态: Accepted

基本信

源代码

```
n = int(input())
patients = []

for i in range(n):
    patient_info = input().split()
    patient_id = patient_info[0]
    age = int(patient_info[1])
    patients.append((patient_id, age, i))

patients.sort(key=lambda x: (-x[1] >= 60, -x[1] if x[1] >= 60 else x[2])

for patient in patients:
    print(patient[0])
```

是
提交
日
提交

©2002-2022 POJ 京ICP备20010980号-1

E23555: 节省存储的矩阵乘法

implementation, matrices,
<http://cs101.openjudge.cn/practice/23555/>

思路：用 X 存储矩阵的非零元素，用 Y 存储另一矩阵的非零元素，用 $result$ 存储结果矩阵中的非零元素，然后计算矩阵乘法，并按照先行号后列号的方式进行递增排序，最后输出结果即可。（25min）

代码：

```
n, m1, m2 = map(int, input().split())
X = []
for _ in range(m1):
    X.append(tuple(map(int, input().split())))
Y = []
for _ in range(m2):
    Y.append(tuple(map(int, input().split())))
result = []
```

```
for i in range(n):
    for j in range(n):
        sum_ij = 0
        for k in range(n):
            for x in X:
                if x[0] == i and x[1] == k:
                    for y in Y:
                        if y[0] == k and y[1] == j:
                            sum_ij += x[2] * y[2]
        if sum_ij != 0:
            result.append((i, j, sum_ij))
result.sort()
for triple in result:
    print(*triple)
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

基本

源代码

```
n, m1, m2 = map(int, input().split())
X = []
for _ in range(m1):
    X.append(tuple(map(int, input().split())))
Y = []
for _ in range(m2):
    Y.append(tuple(map(int, input().split())))
result = []
for i in range(n):
    for j in range(n):
        sum_ij = 0
        for k in range(n):
            for x in X:
                if x[0] == i and x[1] == k:
                    for y in Y:
                        if y[0] == k and y[1] == j:
                            sum_ij += x[2] * y[2]
        if sum_ij != 0:
            result.append((i, j, sum_ij))
result.sort()
for triple in result:
    print(*triple)
```

提

提

©2002-2022 POJ 京ICP备20010980号-1

M18182: 打怪兽

implementation/sortings/data structures,
<http://cs101.openjudge.cn/practice/18182/>

思路: 核心 1: `data.sort(key = lambda x:(x[0],-x[1]))` 对
`data` 列表中的技能信息进行排序。排序依据是一个由 `lambda` 函数

定义的键，这里按照技能的可使用时刻 `x[0]` 进行升序排序，对于可使用时刻相同的技能，按照能造成的伤害值 `x[1]` 的相反数进行降序排序。

核心 2：然后通过一个循环 `for i in range (n)` 遍历排序后的 `data` 列表，看技能是否被使用

核心 3：更新怪兽血量并判断是否杀死怪兽（2h）

代码：

```
t = int(input())
for _ in range (t):
    n,m,b = map(int,input().split())
    data = []
    for i in range (n):
        t,x = map(int,input().split())
        data.append((t,x))
    num = {}
    can = True
    data.sort(key = lambda x:(x[0],-x[1]))
    for i in range (n):
        if data[i][0] not in num:
            num[data[i][0]] = 0
```

```
num[data[i][0]] += 1

if num[data[i][0]] <= m:

    b = b-data[i][1]

if b <= 0:

    print(data[i][0])

    can = False

    break

if can:

    print("alive")
```

代码运行截图 [<mark>](#)（至少包含有"Accepted"）[</mark>](#)

状态: Accepted

源代码

```
t = int(input())
for _ in range (t):
    n,m,b = map(int,input().split())
    data = []
    for i in range (n):
        t,x = map(int,input().split())
        data.append((t,x))
    num = {}
    can = True
    data.sort(key = lambda x:(x[0],-x[1]))
    for i in range (n):
        if data[i][0] not in num:
            num[data[i][0]] = 0
            num[data[i][0]] += 1
        if num[data[i][0]] <= m:
            b = b-data[i][1]
        if b <= 0:
            print(data[i][0])
            can = False
            break
    if can:
        print("alive")
```

基本信息

#: 47085094
题目: 18182
提交人: EuphoriaJ
内存: 3816kB
时间: 80ms
语言: Python3
提交时间: 2024-11-10 20:25:19

M28780: 零钱兑换 3

dp, <http://cs101.openjudge.cn/practice/28780/>

思路：核心 1：创建一个长度为 $m + 1$ 的列表 `dp`，并将所有元素初始化为正无穷大（`float('inf')`）。这个列表用于存储达到每个金额所需的最少硬币数。

核心 2：利用外层循环（遍历从 1 到 m 的每个金额）与内层循环（判断当前硬币面值是否小于等于当前要凑的金额 i ）

核心 3：检查达到目标金额 m 所需的最少硬币数是否仍然是正无穷大。（1h）

代码：

```
n, m = map(int, input().split())
coins = list(map(int, input().split()))
dp = [float('inf')] * (m + 1)
```

```
dp[0] = 0
for i in range(1, m + 1):
    for coin in coins:
        if coin <= i:
            dp[i] = min(dp[i], dp[i - coin] + 1)
if dp[m] == float('inf'):
    print(-1)
else:
    print(dp[m])
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47084526提交状态

状态: Accepted

源代码

```
n, m = map(int, input().split())
coins = list(map(int, input().split()))
dp = [float('inf')] * (m + 1)
dp[0] = 0
for i in range(1, m + 1):
    for coin in coins:
        if coin <= i:
            dp[i] = min(dp[i], dp[i - coin] + 1)
if dp[m] == float('inf'):
    print(-1)
else:
    print(dp[m])
```

T12757: 阿尔法星人翻译官

implementation,
<http://cs101.openjudge.cn/practice/12757>

思路: 首先定义由字母到数字的映射字典, 然后初始化 `temp`, `result`, `is_negative` 这三个变量, 再遍历输入的英文单词列表并转换, 最后输出结果即可。(45min)

代码:

```
num_words = {  
    "negative": -1,  
    "zero": 0,  
    "one": 1,  
    "two": 2,
```

```
"three": 3,  
"four": 4,  
"five": 5,  
"six": 6,  
"seven": 7,  
"eight": 8,  
"nine": 9,  
"ten": 10,  
"eleven": 11,  
"twelve": 12,  
"thirteen": 13,  
"fourteen": 14,  
"fifteen": 15,  
"sixteen": 16,  
"seventeen": 17,  
"eighteen": 18,  
"nineteen": 19,  
"twenty": 20,  
"thirty": 30,  
"forty": 40,  
"fifty": 50,  
"sixty": 60,
```

```
    "seventy": 70,  
    "eighty": 80,  
    "ninety": 90,  
    "hundred": 100,  
    "thousand": 1000,  
    "million": 1000000  
}
```

```
words = input().split()  
result = 0  
temp = 0  
is_negative = False  
for word in words:  
    if word == "negative":  
        is_negative = True  
    elif word == "million":  
        result += temp * num_words["million"]  
        temp = 0  
    elif word == "thousand":  
        result += temp * num_words["thousand"]  
        temp = 0  
    elif word == "hundred":
```

```
        temp *= 100
    else:
        value = num_words[word]
        if value >= 100:
            temp += value
        else:
            temp += value
result += temp
if is_negative:
    result = -result
print(result)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47084641提交状态

状态: **Accepted**

源代码

```
num_words = {  
    "negative": -1,  
    "zero": 0,  
    "one": 1,  
    "two": 2,  
    "three": 3,  
    "four": 4,  
    "five": 5,  
    "six": 6,  
    "seven": 7,  
    "eight": 8,  
    "nine": 9,  
    "ten": 10,  
    "eleven": 11,
```


T16528: 充实的寒假生活

greedy/dp, cs10117 Final Exam,
<http://cs101.openjudge.cn/practice/16528/>

思路:

核心 1: 使用列表的 `sort` 方法对 `activities` 列表进行排序（通过 `lambda` 函数，即 `lambda x: x[1]`，表示按照元组中的第二个元素（也就是活动的结束时间）进行排序）。

核心 2: 首先初始化 `count` 和 `current_end_time` 两个变量，然后通过一个循环遍历已经按照结束时间排好序的 `activities` 列表。

(30min)

代码:

```
n = int(input())
activities = []
for _ in range(n):
    start, end = map(int, input().split())
    activities.append((start, end))
```

```
activities.sort(key=lambda x: x[1])

count = 0

current_end_time = -1

for activity in activities:

    if activity[0] > current_end_time:

        count += 1

        current_end_time = activity[1]

print(count)
```

代码运行截图 [<mark>](#)（至少包含有"Accepted"）[</mark>](#)

状态: Accepted

源代码

```
n = int(input())
activities = []
for _ in range(n):
    start, end = map(int, input().split())
    activities.append((start, end))
activities.sort(key=lambda x: x[1])
count = 0
current_end_time = -1
for activity in activities:
    if activity[0] > current_end_time:
        count += 1
        current_end_time = activity[1]
print(count)
```

2. 学习总结和收获

<mark>如果作业题目简单，有否额外练习题目，比如：OJ“计概2024fall 每日选做”、CF、LeetCode、洛谷等网站题目。</mark> 又是除了作业外啥都没做的一周（数分的压力太大了。。）（有几个题，比如打怪兽还 WA 了很久），下周二就可以开始补计概了，希望还能跟上.....