

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>

姓名：李彦臻

学号：2300010821

学院：数学科学学院

月考成绩：AC4

****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、数学题目）”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://typora.io.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/>

思路：这道题难度并不大，只需要把病人按照年龄是否大于 60 分类，然后分情况处理即可。不过在具体细节上，因为有点忘记那个对字典排序的方法了，所以稍微浪费了点时间回忆方法，下次希望能更熟练一点~

代码：

```
```python
```

```
```
```

```
n=int(input())
```

```
old_age=[]
old={}
young=[]
for i in range(n):
    data=list(map(str,input().split()))
    a,b=data[0],int(data[1])
    if b >= 60:
        if b in old_age:
            old[b].append(a)
        else:
            old[b]=[a]
            old_age.append(b)
    else:
        young.append(data)

sorted_old=sorted(old.items(),key=lambda item:item[0],reverse=True)
for lists in sorted_old:
    for value in lists[1]:
        print(value)
for lists in young:
    print(lists[0])
```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

 **CS101 / 20241107 cs101 Mock Exam立冬 已经结束**

题目 排名 状态 统计 提问

#47010756提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n=int(input())

old_age=[]
old={}
young=[]

for i in range(n):
    data=list(map(str,input().split()))
    a,b=data[0],int(data[1])
    if b >= 60:
        if b in old_age:
            old[b].append(a)
        else:
            old[b]=[a]
            old_age.append(b)
    else:
        young.append(data)

sorted_old=sorted(old.items(),key=lambda item:item[0],reverse=True)

for lists in sorted_old:
    for value in lists[1]:
        print(value)

for lists in young:
    print(lists[0])
```

基本信息

```
#: 47010756
题目: E07618
提交人: 24n2300010821
内存: 3656kB
时间: 27ms
语言: Python3
提交时间: 2024-11-07 15:39:25
```

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

E23555: 节省存储的矩阵乘法

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

思路：先把矩阵 X 和 Y 初始化为 0 矩阵，接着往里面添加非零元素，再利用矩阵运算规则把 XY 的乘积矩阵 Z 算出来，最后再根据行和列的顺序把 Z 中的非零元素列出来即可~

代码：

```
```python
...

data=list(map(int,input().split()))
n,a,b=data[0],data[1],data[2]

matone=[[0]*n for _ in range(n)]
mattwo=[[0]*n for _ in range(n)]
for i in range(a):
 data=list(map(int,input().split()))
 x,y,z=data[0],data[1],data[2]
 matone[x][y]=z
for i in range(b):
 data=list(map(int,input().split()))
 x,y,z=data[0],data[1],data[2]
 mattwo[x][y]=z

finale=[[0]*n for _ in range(n)]for i in range(n):
 for j in range(n):
 finale[i][j]=sum(matone[i][k]*mattwo[k][j] for k in range(n))
 if finale[i][j] != 0:
 print(f"{i} {j} {finale[i][j]}")
```
```

代码运行截图 ==（至少包含有“Accepted”）==

OpenJudge

题目ID, 标题, 描述

24n2300010821

信箱

账号

CS101 / 20241107 cs101 Mock Exam立冬 已经结束

[题目](#)[排名](#)[状态](#)[统计](#)[提问](#)

#47011223提交状态

[查看](#)[提交](#)[统计](#)[提问](#)

状态: Accepted

源代码

```
data=list(map(int,input().split()))
n,a,b=data[0],data[1],data[2]

matone=[0]*n
mattwo=[0]*n

for i in range(a):
    data=list(map(int,input().split()))
    x,y,z=data[0],data[1],data[2]
    matone[x][y]=z

for i in range(b):
    data=list(map(int,input().split()))
    x,y,z=data[0],data[1],data[2]
    mattwo[x][y]=z

finale=[0]*n
for i in range(n):
    for j in range(n):
        finale[i][j]=sum(matone[i][k]*mattwo[k][j] for k in range(n))
    if finale[i][j] != 0:
        print(f"({i}) ({j}) {finale[i][j]}")
```

基本信息

#: 47011223
题目: E23555
提交人: 24n2300010821
内存: 3708kB
时间: 29ms
语言: Python3
提交时间: 2024-11-07 15:50:50

©2002-2022 POJ 京ICP备20010980号-1

[English](#)[帮助](#)[关于](#)

M18182: 打怪兽

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

思路：这道题目和第五题一样，都是思路不难想到但是稍微有点繁琐的题目（不过当然，远远没有第五题繁琐）；第五题主要是代码长情况多，这题主要是要处理的步骤多。这道题具体的思路就是：第一步，先把每一个时刻能用的技能按照伤害排序，并筛选出伤害最高的 m 个技能保留，合并为一个技能组；第二步，把所有的技能组按照时间排序；第三步，把怪兽每个时间点的血量算出来，小于等于 0 停止即可。

代码：

```
```python
...
def mon():
 data=list(map(int,input().split()))
 a,b,c=data[0],data[1],data[2]
 skills={}
 for j in range(a):
 skill=list(map(int,input().split()))
 t,x=skill[0],skill[1]
 if t in skills:
```

```

 skills[t].append(x)
 else:
 skills[t]=[x]

for stats in skills.items():
 time=stats[0]
 lists=stats[1]
 lists.sort(reverse=True)
 if len(lists) > b:
 skills[time] = lists[:b]

finale=sorted(skills.items(),key=lambda item:item[0])

status=True
for move in finale:
 damage=sum(move[1])
 c -= damage
 if c <= 0:
 status=False
 print(move[0])
 break

if status == True:
 print("alive")

n=int(input())
for i in range(n):
 mon()

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

OpenJudge

题目ID, 标题, 描述

24n2300010821

信箱

账号

 **CS101 / 20241107 cs101 Mock Exam立冬 已经结束**

题目 排名 状态 统计 提问

#47012307提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def mon():
 data=list(map(int,input().split()))
 a,b,c=data[0],data[1],data[2]
 skills={}
 for j in range(a):
 skill=list(map(int,input().split()))
 t,x=skill[0],skill[1]
 if t in skills:
 skills[t].append(x)
 else:
 skills[t]=[x]

 for stats in skills.items():
 time=stats[0]
 lists=stats[1]
 lists.sort(reverse=True)
 if len(lists) > b:
 skills[time] = lists[:b]

 finale=sorted(skills.items(),key=lambda item:item[0])

 status=True
 for move in finale:
 damage=sum(move[1])
 c -= damage
 if c <= 0:
```

基本信息

#: 47012307

题目: M18182

提交人: 24n2300010821

内存: 3832kB

时间: 83ms

语言: Python3

提交时间: 2024-11-07 16:12:45

### ### M28780: 零钱兑换 3

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

思路：这道题其实很像 codeforces 上那个 cut ribbon 的题目，而且比那道题更简单一些，所以我考场上做的十分快。具体思路就是直接 dp 就好了，把 dp 表格的第 i 个数记录为能凑成 i 元的最少硬币数，然后利用前面的信息递归出第 i+1 项即可。

代码：

```
```python
```

data=list(map(int,input().split()))
n,m=data[0],data[1]

price=list(map(int,input().split()))

dp=[0]*(m+1)for i in range(1,m+1):
 probable=[]

 for j in range(n):
 if i > price[j] and dp[i - price[j]] != 0:
 probable.append(dp[i - price[j]] + 1)
```

```

 elif i == price[j]:
 probable.append(1)

 if probable:
 dp[i]=min(probable)
if dp[m] == 0:
 print(-1)
else:
 print(dp[m])

```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>



The screenshot shows the OpenJudge submission page for problem T12757. The submission status is 'Accepted'. The source code is as follows:

```

data=list(map(int,input().split()))
n,m=data[0],data[1]

price=list(map(int,input().split()))

dp=[0]*(m+1)
for i in range(1,m+1):
 probable=[]

 for j in range(n):
 if i > price[j] and dp[i - price[j]] != 0:
 probable.append(dp[i - price[j]] + 1)
 elif i == price[j]:
 probable.append(1)

 if probable:
 dp[i]=min(probable)

if dp[m] == 0:
 print(-1)
else:
 print(dp[m])

```

The basic information is as follows:

- #: 47013786
- 题目: M28780
- 提交人: 24n2300010821
- 内存: 8632kB
- 时间: 16402ms
- 语言: Python3
- 提交时间: 2024-11-07 16:31:09

### T12757: 阿尔法星人翻译官

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

思路：这道题目本质上并不难，只需要把整个单词划分为若干个部分，第一个部分为 million、第二个部分为 thousand、第三个部分为剩下的尾数，然后每个部分内部讨论即可。其实每个部分内部的讨论是高度相似的，甚至可以直接把上个部分的内容复制粘贴过来，但是有个问题在于复制粘贴会把上个部分的变量名也粘贴过来，还得一个一个改，很烦。。。总之就是一道本质难度不大，但是实际很繁琐的题目。很考验大家的细心程度！

代码：

```

```python
```

sentence=list(map(str,input().split()))
a=0

status=True
if 'negative' in sentence:
 status=False
 sentence.remove('negative')

if 'million' in sentence:
 t=sentence.index('million')
 mill=sentence[:t]
 b=0
 if 'hundred' in mill:
 if mill[0] == 'one':
 b += 100
 elif mill[0] == 'two':
 b += 200
 elif mill[0] == 'three':
 b += 300
 elif mill[0] == 'four':
 b += 400
 elif mill[0] == 'five':
 b += 500
 elif mill[0] == 'six':
 b += 600
 elif mill[0] == 'seven':
 b += 700
 elif mill[0] == 'eight':
 b += 800
 elif mill[0] == 'nine':
 b += 900
 if mill[-1] != 'hundred':
 if mill[-1] == 'one':
 b += 1
 elif mill[-1] == 'two':
 b += 2
 elif mill[-1] == 'three':
 b += 3
 elif mill[-1] == 'four':
 b += 4
 elif mill[-1] == 'five':
 b += 5
 elif mill[-1] == 'six':

```



```

 b += 6
 elif mill[-1] == 'seven':
 b += 7
 elif mill[-1] == 'eight':
 b += 8
 elif mill[-1] == 'nine':
 b += 9
 elif mill[-1] == 'ten':
 b += 10
 elif mill[-1] == 'eleven':
 b += 11
 elif mill[-1] == 'twelve':
 b += 12
 elif mill[-1] == 'thirteen':
 b += 13
 elif mill[-1] == 'fourteen':
 b += 14
 elif mill[-1] == 'fifteen':
 b += 15
 elif mill[-1] == 'sixteen':
 b += 16
 elif mill[-1] == 'seventeen':
 b += 17
 elif mill[-1] == 'eighteen':
 b += 18
 elif mill[-1] == 'nineteen':
 b += 19
 if 'twenty' in mill:
 b += 20
 if 'thirty' in mill:
 b += 30
 if 'forty' in mill:
 b += 40
 if 'fifty' in mill:
 b += 50
 if 'sixty' in mill:
 b += 60
 if 'seventy' in mill:
 b += 70
 if 'eighty' in mill:
 b += 80
 if 'ninety' in mill:
 b += 90
 a += b*1000000
 sentence=sentence[t+1:]

if 'thousand' in sentence:

```

```

t=sentence.index(' thousand')
thou=sentence[:t]
b=0
if 'hundred' in thou:
 if thou[0] == 'one':
 b += 100
 elif thou[0] == 'two':
 b += 200
 elif thou[0] == 'three':
 b += 300
 elif thou[0] == 'four':
 b += 400
 elif thou[0] == 'five':
 b += 500
 elif thou[0] == 'six':
 b += 600
 elif thou[0] == 'seven':
 b += 700
 elif thou[0] == 'eight':
 b += 800
 elif thou[0] == 'nine':
 b += 900
if thou[-1] != 'hundred':
 if thou[-1] == 'one':
 b += 1
 elif thou[-1] == 'two':
 b += 2
 elif thou[-1] == 'three':
 b += 3
 elif thou[-1] == 'four':
 b += 4
 elif thou[-1] == 'five':
 b += 5
 elif thou[-1] == 'six':
 b += 6
 elif thou[-1] == 'seven':
 b += 7
 elif thou[-1] == 'eight':
 b += 8
 elif thou[-1] == 'nine':
 b += 9
 elif thou[-1] == 'ten':
 b += 10
 elif thou[-1] == 'eleven':
 b += 11
 elif thou[-1] == 'twelve':
 b += 12

```

```

 elif thou[-1] == 'thirteen':
 b += 13
 elif thou[-1] == 'fourteen':
 b += 14
 elif thou[-1] == 'fifteen':
 b += 15
 elif thou[-1] == 'sixteen':
 b += 16
 elif thou[-1] == 'seventeen':
 b += 17
 elif thou[-1] == 'eighteen':
 b += 18
 elif thou[-1] == 'nineteen':
 b += 19
 if 'twenty' in thou:
 b += 20
 if 'thirty' in thou:
 b += 30
 if 'forty' in thou:
 b += 40
 if 'fifty' in thou:
 b += 50
 if 'sixty' in thou:
 b += 60
 if 'seventy' in thou:
 b += 70
 if 'eighty' in thou:
 b += 80
 if 'ninety' in thou:
 b += 90
 a += b*1000
 sentence=sentence[t+1:]

b=0
if len(sentence) == 0:
 a += b
else:
 if 'hundred' in sentence:
 if sentence[0] == 'one':
 b += 100
 elif sentence[0] == 'two':
 b += 200
 elif sentence[0] == 'three':
 b += 300
 elif sentence[0] == 'four':
 b += 400
 elif sentence[0] == 'five':

```

```
 b += 500
 elif sentence[0] == 'six':
 b += 600
 elif sentence[0] == 'seven':
 b += 700
 elif sentence[0] == 'eight':
 b += 800
 elif sentence[0] == 'nine':
 b += 900
if sentence[-1] != 'hundred':
 if sentence[-1] == 'one':
 b += 1
 elif sentence[-1] == 'two':
 b += 2
 elif sentence[-1] == 'three':
 b += 3
 elif sentence[-1] == 'four':
 b += 4
 elif sentence[-1] == 'five':
 b += 5
 elif sentence[-1] == 'six':
 b += 6
 elif sentence[-1] == 'seven':
 b += 7
 elif sentence[-1] == 'eight':
 b += 8
 elif sentence[-1] == 'nine':
 b += 9
 elif sentence[-1] == 'ten':
 b += 10
 elif sentence[-1] == 'eleven':
 b += 11
 elif sentence[-1] == 'twelve':
 b += 12
 elif sentence[-1] == 'thirteen':
 b += 13
 elif sentence[-1] == 'fourteen':
 b += 14
 elif sentence[-1] == 'fifteen':
 b += 15
 elif sentence[-1] == 'sixteen':
 b += 16
 elif sentence[-1] == 'seventeen':
 b += 17
 elif sentence[-1] == 'eighteen':
 b += 18
 elif sentence[-1] == 'nineteen':
```

```
 b += 19
 if 'twenty' in sentence:
 b += 20
 if 'thirty' in sentence:
 b += 30
 if 'forty' in sentence:
 b += 40
 if 'fifty' in sentence:
 b += 50
 if 'sixty' in sentence:
 b += 60
 if 'seventy' in sentence:
 b += 70
 if 'eighty' in sentence:
 b += 80
 if 'ninety' in sentence:
 b += 90
 a += b

if status == False:
 a=-a
print(a)
```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

OpenJudge

题目ID, 标题, 描述

24n2300010821

信箱

账号

CS101 / 计概2023fall每日选做

题目

排名

状态

提问

#47122210提交状态

查看

提交

统计

提问

状态: Accepted

源代码

pre>
sentence=list(map(str,input().split()))
a=0

status=True
if 'negative' in sentence:
 status=False
 sentence.remove('negative')

if 'million' in sentence:
 t=sentence.index('million')
 mill=sentence[:t]
 b=0
 if 'hundred' in mill:
 if mill[0] == 'one':
 b += 100
 elif mill[0] == 'two':
 b += 200
 elif mill[0] == 'three':
 b += 300
 elif mill[0] == 'four':
 b += 400
 elif mill[0] == 'five':
 b += 500
 elif mill[0] == 'six':
 b += 600

基本信息

#: 47122210

题目: 12757

提交人: 24n2300010821

内存: 4280kB

时间: 28ms

语言: Python3

提交时间: 2024-11-12 21:05:55

### ### T16528: 充实的寒假生活

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

思路：这道题目其实只要想到了思路是不难的，至少比第四题稍微简单一点。  
主要的思想就是每次选活动的时候，都应尽量让此次的活动结束时间靠前，这样就能给接下来的活动预留尽量多的时间；并且不难想象，这样显然是最优策略！  
例如，选第一个活动时，就选那个结束时间最靠前的；接下来，把所有无法容纳进日程的活动都排除掉，再在剩下的活动里选结束时间最靠前的作为第二个活动，以此类推~

代码：

```
```python
...
n=int(input())
events=[]
for i in range(n):
    data=list(map(int,input().split()))
    events.append(data)

sorted_events=sorted(events,key=lambda item:item[1])

good_events=[]
while sorted_events:
    good_events.append(sorted_events[0])
    end_day=sorted_events[0][1]
    sorted_events=[event for event in sorted_events if event[0] > end_day]

print(len(good_events))
```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

OpenJudge

题目ID, 标题, 描述

24n2300010821

信箱

账号

CS101 / 计概2023fall每日选做

题目

排名

状态

提问

#47122985提交状态

查看

提交

统计

提问

状态: Accepted

源代码

```
n=int(input())
events=[]
for i in range(n):
    data=list(map(int,input().split()))
    events.append(data)

sorted_events=sorted(events,key=lambda item:item[1])

good_events=[]
while sorted_events:
    good_events.append(sorted_events[0])
    end_day=sorted_events[0][1]
    sorted_events=[event for event in sorted_events if event[0] > end_d

print(len(good_events))
```

基本信息

#: 47122985

题目: 16528

提交人: 24n2300010821

内存: 4104kB

时间: 35ms

语言: Python3

提交时间: 2024-11-12 21:42:25

©2002-2022 POJ 京ICP备20010980号-1

English

帮助

关于

2. 学习总结和收获

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。

这一次考试我整体来讲做的还是比较顺利的,但是因为这次考试的时间与我的一个专业课冲突了,所以稍微到晚了一点点;再加上这是第一次来月考,对设备不太熟悉,花了好几分钟才进入了 openjudge 的网站,导致我考试的有效时间比其他同学少了十几分钟 hhh。不过,至少这一次来了给了我不少经验,下次正式考试的时候应该就不会出现这样的情况了~

对于具体的考试情况而言,我第一题因为不太记得对字典的排序方法了,所以浪费了一点时间 qwq。第二题、第三题、第四题都做的比较顺利,没有花太多时间;尤其是对于第四题而言,虽然根据同学们的说法而言,这道题较难(几乎是最难的一道),但是我做的 dp 题目比较多,所以这道题做起来出乎意料得快,还是蛮开心的~

可惜的是,做第五题的时候,因为太过于繁琐了,所以浪费了很多时间在修改变量名和一些其他细节上,导致我最后终于把代码调试完了上传的时候正好已经 5 点整,上传不上去了,有点可惜,没有 AC5。而我考完之后才发现第六题其实是一个水题,考场上没看也有点可惜...

另外,在做第六题的过程中我又犯了一个老犯的错误:不能在遍历列表的过程中修改列表!!(不要在 for 循环中 remove 或者 append! 这样很容易导致索引错误!!)下次一定要引以为鉴~