

Assignment #7: Nov Mock Exam 立冬

Updated 1646 GMT+8 Nov 7, 2024

2024 fall, Compiled by <mark>汤伟杰，信息管理系</mark>

说明：

- 1) 月考：AC4<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

sortings, <http://cs101.openjudge.cn/practice/07618/>

思路：

把老人和年轻人用两个列表储存，如果不确定 sort 是不是稳定的可以自行编入一个数据然后用 sort 输出看看顺序会不会变。当时考试的时候测试了一下发现没问题，才放心用。有一个坑点是 b 要提前转成 int，否则后续排序的时候会按照 b 的字典序而不是大小来排序（当时 debug 半天才发现问题）

代码：

```
t=int(input())
old,young=[],[]
for _ in range(t):
    a,b=input().split()
    b=int(b)
    if b>=60:
        old.append([a,b])
    else:
        young.append([a,b])
old.sort(key=lambda x:x[1],reverse=True)
for i in old:
    print(i[0])
for i in young:
    print(i[0])
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47015340提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
t=int(input())
old,young=[],[]
for _ in range(t):
    a,b=input().split()
    b=int(b)
    if b>=60:
        old.append([a,b])
    else:
        young.append([a,b])
old.sort(key=lambda x:x[1],reverse=True)
for i in old:
    print(i[0])
for i in young:
    print(i[0])
```

基本信息

#: 47015340
题目: E07618
提交人: 24n2400016635
内存: 3608kB
时间: 30ms
语言: Python3
提交时间: 2024-11-07 16:49:46

©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:

由于以前做过类似的题目，所以这个题直接把题目输入放到矩阵里面就好了，在计算乘法之后一旦有值不是 0 就可以在遍历中直接把位置和值输出。

代码:

```
n,m1,m2=map(int,input().split())
a=[[0]*n for _ in range(n)]
b=[[0]*n for _ in range(n)]
for _ in range(m1):
    x,y,v=map(int,input().split())
    a[x][y]=v
for _ in range(m2):
    x,y,v=map(int,input().split())
    b[x][y]=v
c=[[0]*n for _ in range(n)]
for i in range(n):
    for j in range(n):
        c[i][j]=sum(a[i][k]*b[k][j] for k in range(n))
        if c[i][j]!=0:
            print(i,j,c[i][j])
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
n,m1,m2=map(int,input().split())
a=[[0]*n for _ in range(n)]
b=[[0]*n for _ in range(n)]
for _ in range(m1):
    x,y,v=map(int,input().split())
    a[x][y]=v
for _ in range(m2):
    x,y,v=map(int,input().split())
    b[x][y]=v
c=[[0]*n for _ in range(n)]
for i in range(n):
    for j in range(n):
        c[i][j]=sum(a[i][k]*b[k][j] for k in range(n))
        if c[i][j]!=0:
            print(i,j,c[i][j])
```

基本信息

#: 47010489
题目: E23555
提交人: 24n2400016635
内存: 3776kB
时间: 29ms
语言: Python3
提交时间: 2024-11-07 15:33:00

M18182: 打怪兽

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

思路:

由于有 m 的限制，每次放技能要取 $\min(m, \text{技能的个数})$ 来防止越界。所以在输入的时候可以把每个时刻的技能数用一个字典存储起来。之后就是贪心。中间使用 `sort(key=lambda x:(x[0],-x[1]))` 来按照第一个元素升序，第二个元素降序。

代码:

```
t=int(input())
for _ in range(t):
    skill=[];num={}
    n,m,b=map(int,input().split())
    for _ in range(n):
        t,x=map(int,input().split())
        skill.append([t,x])
        if t not in num:
            num[t]=1
        else:
            num[t]+=1
    skill.sort(key=lambda x:(x[0],-x[1]))
    # print(skill)
    i=0;f=False;ans=0
    while i<n:
        curr_time=skill[i][0]
        for j in range(min(m,num[curr_time])):
            b-=skill[i+j][1]
            i=i+num[curr_time]
            if b<=0:
                ans=curr_time
                f=True
                break
    if not f:
        print('alive')
    else:
```

```
print(ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47011375提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
t=int(input())
for _ in range(t):
    skill=[];num={}
    n,m,b=map(int,input().split())
    for _ in range(n):
        t,x=map(int,input().split())
        skill.append([t,x])
        if t not in num:
            num[t]=1
        else:
            num[t]+=1
    skill.sort(key=lambda x:(x[0],-x[1]))
    # print(skill)
    i=0;f=False;ans=0
    while i<n:
        curr_time=skill[i][0]
        for j in range(min(m,num[curr_time])):
            b-=skill[i+j][1]
            i=i+num[curr_time]
            if b<=0:
                ans=curr_time
                f=True
                break
    if not f:
        print('alive')
    else:
        print(ans)
```

基本信息

#: 47011375

题目: M18182

提交人: 24n2400016635

内存: 3808kB

时间: 80ms

语言: Python3

提交时间: 2024-11-07 15:54:41

M28780: 零钱兑换 3

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

思路:

由于可能存在找不到的情况，所以设定每一个 dp 为正无穷（除了第一个是 0，因为需要 0 个硬币就可以找零 0 元）。这样看来每一个 dp 都是依赖于 min 的第二项，能在先前的状态找到答案的直接拿来用再+1，找不到就还是 inf 的状态，然后查看 dp[-1]是不是 inf 来输出。

代码:

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

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

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

基本信息

#: 47091541
题目: 28780
提交人: 24n2400016635
内存: 13532kB
时间: 14564ms
语言: Python3
提交时间: 2024-11-11 11:08:42

T12757: 阿尔法星人翻译官

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

思路:

考试的时候发现了 thousand 和 million 和 hundred 的特殊之处，但是选择的是从前往后遍历每个词，逻辑没有捋清楚，导致耗时很长且一直 WA。考完参考了群里焦炜宸同学的代码之后，感觉思路清晰重写了一遍，群里的的代码真的牛！

代码:

```
t={'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}
b={'hundred':100, 'thousand':1000, 'million':10**6}

def transfer(s):
    if 'million' in s:
        i=s.index('million')
        return transfer(s[:i])*(10**6)+transfer(s[i+1:]) #使用递归，很清晰的思路
    if 'thousand' in s:
        j=s.index('thousand')
        return transfer(s[:j])*1000+transfer(s[j+1:])
    if 'hundred' in s:
        k=s.index('hundred')
        cnt=t[s[0]]*100
        for i in range(2,len(s)):
            cnt+=t[s[i]]
        return cnt

    cnt=0
    for i in range(len(s)):
        cnt+=t[s[i]]
    return cnt

s=list(input().split())
```

```
n=len(s)
if s[0]=='negative':
    ans=-transfer(s[1:])
else:
    ans=transfer(s)
print(ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

（没找到提交入口，不过输入了一个用来 debug 我自己写的错误代码的测试，答案是对的：
one hundred eleven million one hundred eleven thousand one hundred eleven

T16528: 充实的寒假生活

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

思路：

这就是区间问题，按照结束时间排序，然后看找下一个起始时间不超过结束时间的区间，一直贪心。

代码：

```
n=int(input())
t=[]
for _ in range(n):
    a,b=map(int,input().split())
    t.append((a,b))
t.sort(key=lambda x:x[1])
ans,i,j=0,0,0
while j<n and i<n:
    end=t[i][1]
    while j<n and t[j][0]<=end:
        j+=1
    ans+=1
    i=j
print(ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: **Accepted**

源代码

```
n=int(input())
t=[]
for _ in range(n):
    a,b=map(int,input().split())
    t.append((a,b))
t.sort(key=lambda x:x[1])
ans,i,j=0,0,0
while j<n and i<n:
    end=t[i][1]
    while j<n and t[j][0]<=end:
        j+=1
    ans+=1
    i=j
print(ans)
```

基本信息

#: 47014648
题目: T16528
提交人: 24n2400016635
内存: 5312kB
时间: 33ms
语言: Python3
提交时间: 2024-11-07 16:41:17

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>

月考这一周每天都在做高数感觉自己快窒息了，考试的时候在上专业课不过抢到了教室角落完全不影响月考做题（，最繁琐的是翻译题，需要提前想好怎么转化，考试的时候脑子一团乱麻，总是导致出错。考完看看群里老师发的代码，在众多代码中发现焦玮宸同学这道题是用递归写法，很清晰的思路；还有矩阵那道题，群里很多是用字典存储，而我当时就只想到创建 3 个二维矩阵一个一个填充并计算。

周日考完高数，目前每日选做落下许多，开始往前追。