**Assignment #7: Nov Mock Exam立冬**

2024 fall, Complied by 吕金浩，物理学院

⽉考： AC6（喜

**1. 题目**

**E07618: 病人排队**

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

思路：把循环变量也存储下来，以保证排序稳定性

代码：

n=int(input())

old=[]

young=[]

for i in range(n):

a=input().split()

iden=a[0]

years=int(a[1])

if years>=60:

old.append((-years, i, iden))

else:

young.append(iden)

old.sort()

for x in old:

print(x[-1])

for x in young:

print(x)



**E23555: 节省存储的矩阵乘法**

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

代码：

n,m1,m2=map(int,input().split())

matrix1=[[0]\*n for \_ in range(n)]

matrix2=[[0]\*n for \_ in range(n)]

for \_ in range(m1):

r,c,v=map(int,input().split())

matrix1[r][c]=v

for \_ in range(m2):

r,c,v=map(int,input().split())

matrix2[r][c]=v

matrix3=[[0]\*n for \_ in range(n)]

for i in range(n):

for j in range(n):

v=0

for k in range(n):

v+=matrix1[i][k]\*matrix2[k][j]

if v!=0:

print(str(i)+' '+str(j)+ ' '+str(v))



**M18182: 打怪兽**

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

代码：

for \_ in range(int(input())):

n,m,b=map(int,input().split())

ejineng={}

for \_ in range(n):

t,x=map(int,input().split())

if t not in ejineng:

ejineng[t]=[x]

else:

ejineng[t].append(x)

if\_die=False

a=[x for x in ejineng.keys()]

a.sort()

#print(ejineng)

for t in a:

if len(ejineng[t])<=m:

b-=sum(ejineng[t])

else:

ejineng[t].sort(reverse=True)

b-=sum(ejineng[t][:m])

if b<=0:

if\_die=True

print(t)

break

if not if\_die:

print('alive')



**M28780: 零钱兑换3**

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

代码：

n,total\_num=map(int,input().split())

kinds=[int(x) for x in input().split()]

min\_num=[0]+[float('inf')]\*total\_num

for i in range(1,total\_num+1):

for kind in kinds:

if i>=kind:

min\_num[i]=min(min\_num[i],min\_num[i-kind]+1)

if min\_num[total\_num]<10\*\*8:

print(min\_num[-1])

else:

print(-1)



**T12757: 阿尔法星人翻译官**

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

代码：

ori\_str=input()

words=ori\_str.split()

minus=False

if words[0]=='negative':

minus=True

words=words[1:]

positive\_str=' '.join(words)

#print(positive\_str)

less\_hundred\_value={}

zeroto20=['zero', 'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten', 'eleven', 'twelve', 'thirteen', 'fourteen', 'fifteen', 'sixteen', 'seventeen', 'eighteen', 'nineteen', 'twenty']

twto90=[0,0,'twenty', 'thirty', 'forty', 'fifty', 'sixty', 'seventy', 'eighty', 'ninety']

for i in range(100):

if i<=20:

less\_hundred\_value[i]=zeroto20[i]

else:

a = i // 10

if i%10==0:

less\_hundred\_value[i] = twto90[a]

else:

less\_hundred\_value[i] = twto90[a] + ' '+zeroto20[i%10]

less\_thousand\_value={}

for i in range(1000):

if i<100:

less\_thousand\_value[i]=less\_hundred\_value[i]

else:

a = i // 100

if i % 100 == 0:

less\_thousand\_value[i]=zeroto20[a]+' hundred'

else:

less\_thousand\_value[i] = zeroto20[a] + ' hundred'+' '+less\_hundred\_value[i%100]

strtonum={}

#print(less\_thousand\_value)

for x in less\_thousand\_value:

strtonum[less\_thousand\_value[x]]=x

ans=0

strtonum['']=0

def ttom(x):

if 'thousand' not in x:

return strtonum[x]

else:

a=x.split('thousand')

if len(a)==1:

return strtonum[a[0].strip()]\*1000

else:

return strtonum[a[0].strip()]\*1000+strtonum[a[1].strip()]

if 'million' not in positive\_str:

ans=ttom(positive\_str)

else:

b=positive\_str.split('million')

if len(b)==1:

ans=strtonum[b[0].strip()]\*1000000

else:

ans=strtonum[b[0].strip()]\*1000000+ttom(b[1].strip())

if minus:

ans\*=(-1)

print(ans)



**T16528: 充实的寒假生活**

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

思路：

递归。

首先，对于所有的活动开始时间，只保留下结束时间最早的那一个活动即可接着，考虑从t0时刻开始如何参加最多活动f(t0)：如果t0时刻没有活动开始，就等待到下一次活动开始的时间进行考虑。对于t0时刻开始的活动，若参加，则最多参加活动数f(t0)等于1加上f(t+1)，t是这一活动结束时间；若不参加，则f(t0)=f(t0+1)。即有f(t0)=max(f(t+1)+1,f(t0+1))。

最后，加上lru\_cache之前会TLE，加上之后就31msAC了。

代码（前半部分有点冗杂）：

n=int(input())

start\_end={}

for \_ in range(n):

s,t=map(int,input().split())

if s in start\_end:

start\_end[s]=min(t,start\_end[s])

else:

start\_end[s]=t

#start\_end.append([s,t])

start\_and\_end=[]

for s in start\_end:

if start\_end[s]<=60:

start\_and\_end.append([s, start\_end[s]])

start\_and\_end.sort()

start\_end=start\_and\_end[:]

#print(start\_end)

starts=[x[0] for x in start\_end]

from bisect import \*

import sys

from functools import lru\_cache

@lru\_cache(maxsize=None)

def max\_num(start):

idx=bisect\_left(starts,start)

if idx>=len(start\_end):

return 0

else:

for x in start\_end[idx:]:

return max(max\_num(x[0] + 1), 1 + max\_num(x[1]+1))

#以上两行是我考试时的写法，数据量小不会超时

#按照思路里的说法，上面这两行可以改成下面这样，会更省时：

#return max(max\_num(start\_end[idx][0] + 1), 1 + max\_num(start\_end[idx][1]+1))

#for i in range(61):

print(max\_num(0))



**2. 学习总结和收获**

作为基本零基础的同学，这次能AK还是比较开心的，虽然题目看起来并没有那么困难。

模板题目做起来感觉挺快的，主要时间花在了打怪兽和英文翻译，打怪兽WA几次是因为不太熟悉字典存储方式，不过捣鼓两下还是能弄出来；英文翻译做了一个小时，小错误出现在各种地方，感觉这种题目要尽量避免写矢山以方便排查。

这次月考给我带来了很大信心，希望下次月考和期末考能继续保持。