**Assignment #D: 十全十美**

2024 fall, Complied by 吕金浩，物理学院

**1. 题目**

**02692: 假币问题**

brute force, <http://cs101.openjudge.cn/practice/02692>

思路：主打一个枚举。我在九月份做了这个题目，所以代码显得非常长。

代码：

coins=['A','B','C','D','E','F','G','H','I','J','K','L']  
n=int(input())  
iffeit=True  
for \_ in range(n):  
  
 test1 = [str(x) for x in input().split()]  
 test2 = [str(x) for x in input().split()]  
 test3 = [str(x) for x in input().split()]  
 feitcoin=0  
 weight=''  
 while True:  
  
 suppose=coins[feitcoin]  
 iffeit = True  
  
 if (test1[0].find(suppose)!=-1 and test1[2]!='up') or (test1[1].find(suppose)!=-1 and test1[2]!='down'):  
 iffeit=False  
 if test1[0].find(suppose)==-1 and test1[1].find(suppose)==-1 and test1[2]!='even':  
 iffeit=False  
 if (test2[0].find(suppose)!=-1 and test2[2]!='up') or (test2[1].find(suppose)!=-1 and test2[2]!='down'):  
 iffeit=False  
 if test2[0].find(suppose)==-1 and test2[1].find(suppose)==-1 and test2[2]!='even':  
 iffeit=False  
 if (test3[0].find(suppose)!=-1 and test3[2]!='up') or (test3[1].find(suppose)!=-1 and test3[2]!='down'):  
 iffeit=False  
 if test3[0].find(suppose)==-1 and test3[1].find(suppose)==-1 and test3[2]!='even':  
 iffeit=False  
 if iffeit:  
 weight='heavy'  
 break  
 iffeit = True  
  
 if (test1[0].find(suppose) != -1 and test1[2] != 'down') or (test1[1].find(suppose) != -1 and test1[2] != 'up'):  
 iffeit = False  
 if test1[0].find(suppose)==-1 and test1[1].find(suppose)==-1 and test1[2]!='even':  
 iffeit = False  
 if (test2[0].find(suppose) != -1 and test2[2] != 'down') or (test2[1].find(suppose) != -1 and test2[2] != 'up'):  
 iffeit = False  
 if test2[0].find(suppose)==-1 and test2[1].find(suppose)==-1 and test2[2]!='even':  
 iffeit = False  
 if (test3[0].find(suppose) != -1 and test3[2] != 'down') or (test3[1].find(suppose) != -1 and test3[2] != 'up'):  
 iffeit = False  
 if test3[0].find(suppose)==-1 and test3[1].find(suppose)==-1 and test3[2]!='even':  
 iffeit = False  
 if iffeit:  
 weight = 'light'  
 break  
  
 feitcoin+=1  
 print(str(coins[feitcoin])+' is the counterfeit coin and it is '+weight+'.')



**01088: 滑雪**

dp, dfs similar, <http://cs101.openjudge.cn/practice/01088>

思路：使用递归避免基础情况讨论，利用lru\_cache

代码：

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

matrix=[[int(x) for x in input().split()] for \_ in range(r)]

dx=[0,0,1,-1]

dy=[1,-1,0,0]

res=0

import sys

sys.setrecursionlimit(1<<30)

from functools import lru\_cache

@lru\_cache(maxsize=None)

def dfs(x,y):

ans=1

for i in range(4):

nx=x+dx[i]

ny=y+dy[i]

if 0<=nx<r and 0<=ny<c:

if matrix[nx][ny]<matrix[x][y]:

ans=max(ans,1+dfs(nx,ny))

return ans

for j in range(r):

for k in range(c):

res=max(res,dfs(j,k))

print(res)



**25572: 螃蟹采蘑菇**

bfs, dfs, <http://cs101.openjudge.cn/practice/25572/>

思路：将更左边或更上面的5记为“头”，则螃蟹的位置可以用头的位置来表征。螃蟹的取向可用两个变量hor和ver的True或False来表征。

代码：

n=int(input())

matrix=[[int(x) for x in input().split()] for \_ in range(n) ]

def head():

for i in range(n):

for j in range(n):

if matrix[i][j]==5:

matrix[i][j] = 0

return (i,j)

hdx,hdy=head()

tlx,tly=head()

def fd():

for i in range(n):

for j in range(n):

if matrix[i][j]==9:

matrix[i][j]=0

return (i,j)

tx,ty=fd()

from collections import deque

q=deque()

q.append((hdx,hdy))

inq={(hdx,hdy)}

hor=hdx==tlx

ver=hdy==tly

dx=[1,-1,0,0]

dy=[0,0,1,-1]

while q:

x,y=q.popleft()

if (x==tx and y==ty ) or (hor and x==tx and y+1==ty) or (ver and y==ty and x+1==tx):

print('yes')

break

for k in range(4):

nx=x+dx[k];ny=y+dy[k]

if 0<=nx<n and 0<=ny<n and ((hor and 0<=ny+1<n) or (ver and 0<=nx+1<n)) and (nx,ny) not in inq:

if matrix[nx][ny]!=1 and ((hor and matrix[nx][ny+1]!=1) or (ver and matrix[nx+1][ny]!=1)):

q.append((nx,ny))

inq.add((nx,ny))

else:

print('no')



**27373: 最大整数**

dp, <http://cs101.openjudge.cn/practice/27373/>

思路：不看答案真想不到可以起手进行冒泡排序，省去之后关于数字顺序的一系列讨论……还是递归+lru\_cache

代码：

m=int(input());n=int(input())

nums=input().split()

ok=False

import sys

sys.setrecursionlimit(1<<30)

from functools import lru\_cache

while not ok:

ok=True

for i in range(n-1):

if nums[i]+nums[i+1]>nums[i+1]+nums[i]:

nums[i],nums[i+1]=nums[i+1],nums[i]

ok=False

@lru\_cache(maxsize=None)

def mx(x,dig):

if dig<=0:

return ''

if x==0:

if len(nums[0])<=dig:

return nums[0]

else:

return ''

ans1=mx(x-1,dig)

ans2=''

if dig>=len(nums[x]):

ans2=nums[x]+mx(x-1,dig-len(nums[x]))

else:

return ans1

if not ans1:

return ans2

if not ans2:

return ans1

if int(ans1)<int(ans2):

return ans2

else:

return ans1

print(mx(n-1,m))



**02811: 熄灯问题**

brute force, <http://cs101.openjudge.cn/practice/02811>

思路：也并没有想到去枚举第一行的操作方式……这道题的深拷贝卡了我一会，吸取教训了。计概A的某个班最近的一次测试考到了基本一样的问题，考的是给定某一初态求最少需要关灯数量。

代码：

init=[[int(x) for x in input().split()] for \_ in range(5)]  
  
cur=[]  
dm=[1,-1,0,0]  
dn=[0,0,1,-1]  
def turnoff(m,n):  
 cur[m][n]=1-cur[m][n]  
 for i in range(4):  
 nm=m+dm[i]  
 nn=n+dn[i]  
 if 0<=nm<5 and 0<=nn<6:  
 cur[nm][nn]=1-cur[nm][nn]  
ans=[]  
  
for y in range(64):  
 x=y  
 x=bin(x)[2:]  
 x='0'\*(6-len(x))+x  
 cur=[z[:] for z in init]  
  
 ope = [[0] \* 6 for \_ in range(5)]  
 ope[0]=[int(i) for i in x][:]  
  
 for k in range(6):  
 if ope[0][k]:turnoff(0,k)  
 for j in range(1,5):  
 for k in range(6):  
 if cur[j-1][k]:  
 ope[j][k]=1  
 turnoff(j,k)  
  
 if cur[-1]==[0,0,0,0,0,0]:  
 ans=ope[:]  
 break  
#print(ans)  
for i in range(5):  
 print(' '.join(str(x) for x in ans[i]))



**08210: 河中跳房子**

binary search, greedy, <http://cs101.openjudge.cn/practice/08210/>

思路：做了Aggressive cows，再看到提示的binary search，较为轻松地做出来了。

代码：

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

rocks=[0]

for \_ in range(n):

rocks.append(int(input()))

rocks.append(l)

def valid(x):

cur=0

cnt=0

for i in range(1,n+2):

rock=rocks[i]

if rock-cur<x:

cnt+=1

else:

cur=rock

return cnt<=m

left=1

right=l

while left<right:

mid=(left+right)//2

if valid(mid):

left=mid+1

else:

right=mid

print(left-1)



**2. 学习总结和收获**

最大最小整数似乎是去年的期末题，没做出来，感觉非常慌张……搜索的题目感觉还好，基本不太会卡题，但dp和greedy相关的题目是真没那么好想，感觉非常慌张，乞求老师期末别把greedy放到tough球球了