**Assignment #C: 五味杂陈**

2024 fall, Complied by 吕金浩，物理学院

**1. 题目**

**1115. 取石子游戏**

dfs, <https://www.acwing.com/problem/content/description/1117/>

思路：看到提示后就直接秒了

代码：

def win(x,y):  
 a=max(x,y)  
 b=min(x,y)  
 if a%b==0:  
 return True  
 if a//b>=2:  
 return True  
 return not win(a-b,b)  
while 114514:  
 c,d=map(int,input().split())  
 if c==0:  
 break  
 print('win' if win(c,d) else 'lose')



**25570: 洋葱**

Matrices, <http://cs101.openjudge.cn/practice/25570>

思路：对于奇数层的情况，把中心单独拎出来讨论

代码：

n=int(input())  
matrix=[[int(x) for x in input().split()] for \_ in range(n)]  
def slice\_sum(x):  
 #x代表层数，从第1层开始  
 s1=sum([matrix[x-1][i] for i in range(x-1,n-x+1)])  
 s2=sum([matrix[-x][i] for i in range(x-1,n-x+1)])  
 s3=sum([matrix[i][x-1] for i in range(x,n-x)])  
 s4=sum([matrix[i][-x] for i in range(x,n-x)])  
 ans=sum([s1,s2,s3,s4])  
 return ans  
res=-float('inf')  
for j in range(1,1+n//2):  
 res=max(res,slice\_sum(j))  
if n%2!=0:  
 res=max(res,matrix[n//2][n//2])  
print(res)



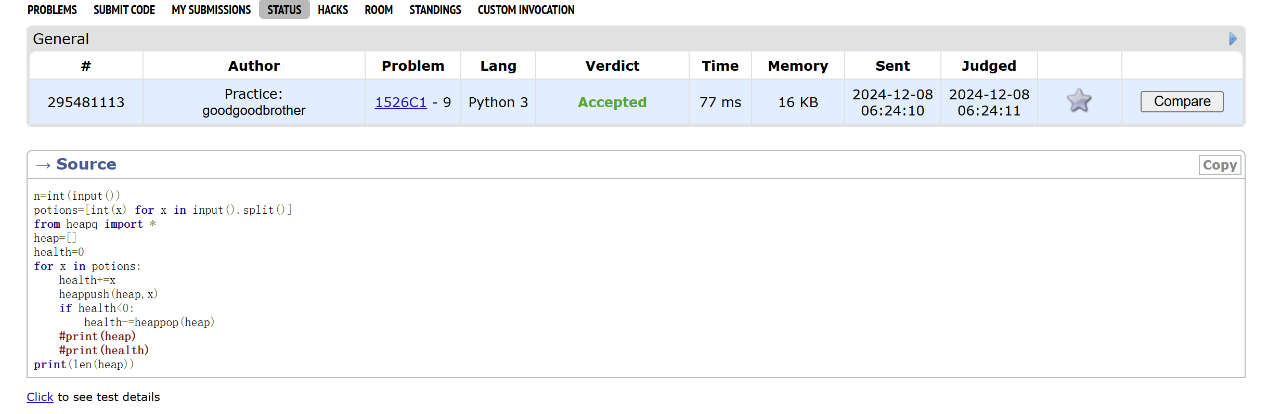
**1526C1. Potions(Easy Version)**

greedy, dp, data structures, brute force, \*1500, <https://codeforces.com/problemset/problem/1526/C1>

思路：自己写了个O(n^2)的TLE了（？？不理解），誊抄了答案。另外答案中if heap这一判断语句多余了。

代码：

n=int(input())  
potions=[int(x) for x in input().split()]  
from heapq import \*  
heap=[]  
health=0  
for x in potions:  
 health+=x  
 heappush(heap,x)  
 if health<0:  
 health-=heappop(heap)  
print(len(heap))



**22067: 快速堆猪**

辅助栈，<http://cs101.openjudge.cn/practice/22067/>

代码：

pigs=[]  
smallpigs=[]  
while True:  
 try:  
 cmd=input()  
 if cmd[1]=='u':  
 push,num=cmd.split()  
 num=int(num)  
 if ( not smallpigs) or num<=smallpigs[-1]:  
 smallpigs.append(num)  
 pigs.append(num)  
 elif pigs:  
 if cmd[1]=='i':  
 print(smallpigs[-1])  
 else:  
 min\_pig=pigs.pop()  
 if min\_pig==smallpigs[-1]:  
 smallpigs.pop()  
  
  
 except EOFError:  
 break



**20106: 走山路**

Dijkstra, <http://cs101.openjudge.cn/practice/20106/>

思路：当走到一个地方时，dfs搜索出所有和它等高的点。逐渐增加其对应的step直到能翻出去为止。

代码：

from collections import deque  
  
dx=[-1,1,0,0]  
dy=[0,0,-1,1]  
  
inf=float('inf')  
  
m,n,p=map(int,input().split())  
matrix=[[inf if x=='#' else int(x) for x in input().split()] for \_ in range(m)]  
  
def bfs(sx,sy,tx,ty):  
 minstep=[[inf]\*n for \_ in range(m)]  
 minstep[sx][sy]=0  
 q=deque()  
 q.append((sx,sy,0))  
 def dfs(a,b):  
 for j in range(4):  
 na = a + dx[j]  
 nb = b + dy[j]  
 if 0 <= na < m and 0 <= nb < n and minstep[na][nb]==inf:  
 if matrix[na][nb]==matrix[a][b]:  
 ms=minstep[a][b]  
 q.append((na, nb, ms))  
 minstep[na][nb] = ms  
 dfs(na,nb)  
 dfs(sx,sy)  
  
 while q :  
 x,y,step=q.popleft()  
 if x==tx and y==ty:  
 return step  
  
 for i in range(4):  
 nx=x+dx[i]  
 ny=y+dy[i]  
  
 if 0<=nx<m and 0<=ny<n and minstep[nx][ny]==inf:  
 if step+1-minstep[x][y]>=abs(matrix[nx][ny]-matrix[x][y]):  
 q.append((nx,ny,step+1))  
 minstep[nx][ny]=step+1  
 dfs(nx,ny)  
 if\_plus = False  
 for i in range(4):  
 nx = x + dx[i]  
 ny = y + dy[i]  
 if 0 <= nx < m and 0 <= ny < n and minstep[nx][ny] == inf and matrix[nx][ny] != inf:  
 if\_plus = True  
 break  
 if if\_plus:  
 q.append((x, y, step + 1))  
  
 return -1  
  
for \_ in range(p):  
 x1,y1,x2,y2=map(int,input().split())  
 if matrix[x1][y1]==inf or matrix[x2][y2]==inf:  
 print('NO')  
 continue  
 res=bfs(x1,y1,x2,y2)  
 print(res if res>=0 else 'NO')



**04129: 变换的迷宫**

bfs, <http://cs101.openjudge.cn/practice/04129/>

思路：inq集合中，记录下走到某一位置时的时间模上k。另外我的答案里面mod这一变量多余了。

代码：

dx=[1,-1,0,0]  
dy=[0,0,1,-1]  
from collections import deque  
  
def bfs(sx,sy,tx,ty):  
 q=deque()  
 q.append((sx,sy,0,0))#x,y,step,mod  
 inq=set()  
 inq.add((sx,sy,0))  
 while q:  
 x,y,step,mod=q.popleft()  
 if x==tx and y==ty:  
 return step  
 for n in range(4):  
 nx=x+dx[n]  
 ny=y+dy[n]  
 if 0<=nx<r and 0<=ny<c and (maze[nx][ny]!='#' or (maze[nx][ny]=='#' and (step+1)%k==0)):  
 if (nx,ny,step%k) not in inq:  
 inq.add((nx,ny,step%k))  
 q.append((nx,ny,step+1,step%k))  
 return -1  
  
for \_ in range(int(input())):  
 r,c,k=map(int,input().split())  
  
 maze=[list(input()) for \_ in range(r)]  
 s,e=0,0  
 for i in range(r):  
 for j in range(c):  
 if maze[i][j]=='S':  
 s=(i,j)  
 if maze[i][j]=='E':  
 e=(i,j)  
 ans=bfs(s[0],s[1],e[0],e[1])  
 print(ans if ans>=0 else 'Oop!')



**2. 学习总结和收获**

做多了bfs感觉并没有那么困难，模板性挺强的。

月考极大地打击了我的自信心，唉，符合作业标题。继续跟进每日选做，期待期末别考太难（和Nov月考差不多就行了球球了）