# Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by <mark>祁黄奕 城市与环境学院</mark>

\*\*说明：\*\*

1）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。

2）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。

3）如果不能在截止前提交作业，请写明原因。

## 1. 题目

### 1115. 取石子游戏

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

思路：数学方法，如果两个堆的石子数满足一定的比例关系，那么先手玩家可以通过正确的策略保证自己一定能够赢得游戏。具体来说，这个比例与黄金分割率有关，即 (1 + sqrt(5)) / 2，通常记作φ。

代码：

```python

while True:  
 a,b = map(int,input().split())  
 if a == b == 0:  
 break  
 if b > a:  
 a,b = b,a  
 if a >= (5\*\*0.5+1)/2\*b or a == b:  
 print('win')  
 else:  
 print('lose')

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



### 25570: 洋葱

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

思路：一层一层的加，用到了螺旋矩阵的方向向量来换方向，用dfs来递归到里面的层数更加方便。

代码：

```python

ans=0  
def dfs(n,s,x,y):  
 global ans  
 if n==1:  
 ans=max(ans,s[x][y])  
 return  
 if n==0:  
 return  
 curr=0  
 d=[(0,1),(1,0),(0,-1),(-1,0)]  
 for i in range(4\*(n-1)):  
 dx,dy=d[(i//(n-1))%4]  
 x+=dx;y+=dy  
 curr+=s[x][y]  
 ans=max(ans,curr)  
 dfs(n-2,s,x+1,y+1)  
  
n=int(input())  
s=[]  
for \_ in range(n):  
 s.append(list(map(int,input().split())))  
dfs(n,s,0,0)  
print(ans)

```

代码运行截图 ==（至少包含有"Accepted"）==



### 1526C1. Potions(Easy Version)

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

思路：遇到喝不下的药水，就看看以前喝下的有没有毒性更强的，要是有就“替换”掉，这样既保证了喝的药水数量不变，又保证了健康数值肯定为正。而查看以前喝下的最毒的药水，用到了heapq来快速查询。

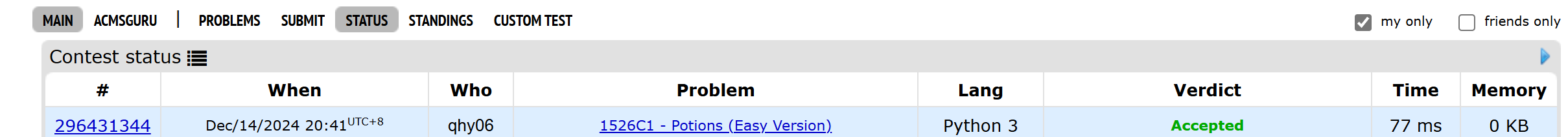
代码：

```python

import heapq  
n=int(input())  
s=list(map(int,input().split()))  
health=0  
drunk=0  
heap=[]  
for p in s:  
 if p+health>=0:  
 drunk+=1  
 heapq.heappush(heap,p)  
 health+=p  
 elif heap and p>heap[0]:  
 smallest=heapq.heappop(heap)  
 health-=smallest  
 heapq.heappush(heap,p)  
 health+=p  
print(drunk)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



### 22067: 快速堆猪

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

思路：

代码：

```python

import sys  
stack = []  
min\_stack = []  
for a in sys.stdin:  
 a = a.strip()  
 if not a:  
 break  
 if a.startswith('push'):  
 \_,n = a.split()  
 n = int(n)  
 stack.append(n)  
 if not min\_stack or n <= min\_stack[-1]:  
 min\_stack.append(n)  
 if a == 'pop' and stack:  
 top = stack.pop()  
 if top == min\_stack[-1]:  
 min\_stack.pop()  
 if a == 'min' and stack:  
 print(min\_stack[-1])

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



### 20106: 走山路

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

思路：

代码：

```python

import heapq  
dx,dy=[0,-1,1,0],[-1,0,0,1]  
def dijkstra(sx,sy,ex,ey):  
 if s[sx][sy]=='#' or s[ex][ey]=='#':  
 return 'NO'  
 q=[]  
 dist=[[float('inf')]\*m for \_ in range(n)]  
 heapq.heappush(q,(0,sx,sy))  
 dist[sx][sy]=0  
 while q:  
 curr,x,y=heapq.heappop(q)  
 if (x,y)==(ex,ey):  
 return curr  
  
 for i in range(4):  
 nx,ny=x+dx[i],y+dy[i]  
 if 0<=nx<n and 0<=ny<m and s[nx][ny]!='#':  
 new=curr+abs(s[x][y]-s[nx][ny])  
 if new<dist[nx][ny]:  
 heapq.heappush(q,(new,nx,ny))  
 dist[nx][ny]=new  
 return 'NO'  
  
n,m,p=map(int,input().split())  
s=[]  
for i in range(n):  
 line=input().split()  
 for j in range(m):  
 if line[j]!='#':  
 line[j]=int(line[j])  
 s.append(line)  
for \_ in range(p):  
 sx,sy,ex,ey=map(int,input().split())  
 print(dijkstra(sx,sy,ex,ey))

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

文本

中度可信度描述已自动生成

### 04129: 变换的迷宫

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

思路：

代码：

```python

import heapq  
t = int(input())  
for \_ in range(t):  
 grid = []  
 r, c, k = map(int, input().split())  
 s\_found = False  
 e\_found = False  
 for i in range(r):  
 line = list(input())  
 for j in range(c):  
 if line[j] == '.':  
 line[j] = 0  
 elif not s\_found and line[j] == 'S':  
 xs = i  
 ys = j  
 s\_found = True  
 line[j] = 0  
 elif not e\_found and line[j] == 'E':  
 xe = i  
 ye = j  
 e\_found = True  
 line[j] = 0  
 else:  
 line[j] = 1  
 grid.append(line)  
  
 directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]  
 heap = [(0, xs, ys)]  
 inqueue = set((0, xs, ys))  
 found = False  
 while heap:  
 d, x, y = heapq.heappop(heap)  
 if (x, y) == (xe, ye):  
 found = True  
 break  
 for dx, dy in directions:  
 nx, ny = x + dx, y + dy  
 if 0 <= nx < r and 0 <= ny < c:  
 nd = d + 1  
 if not nd % k == 0 and grid[nx][ny] == 1:  
 continue  
 if (nd % k, nx, ny) in inqueue:  
 continue  
 heapq.heappush(heap, (nd, nx, ny))  
 inqueue.add((nd % k, nx, ny))  
  
 if found:  
 print(d)  
 else:  
 print('Oop!')

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

图片包含 图形用户界面

描述已自动生成

## 2. 学习总结和收获

依然焦头烂额，试图看懂但挣扎完毕还是不怎么会。作业题目涉及到dfs、bfs这种算法的题期末打算放弃掉的，已经研究不动了。求先生善良！！！求期末机考善良！！！