**# Assignment #C: 五味杂陈**

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by <mark>陶嘉瑞-物理学院</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：

递推下去即可

代码：

```python

def dfs(a,b):

    global n

    a0=max(a,b)

    b0=min(a,b)

    if b==0 or a0//b0>=2 or a0==b0:

        return n

    else:

        n+=1

        a1=b0

        b1=a0-b0

        dfs(a1,b1)

while True:

    a,b=map(int,input().split())

    n=0

    if a==0 and b==0:

        break

    dfs(a,b)

    print('win' if n%2==0 else '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：

把每层算出来即可，再单独计算奇数时的中心

代码：

```python

n=int(input())

matrix=[]

for i in range(n):

    matrix.append(list(map(int,input().split())))

sumn=[]

s=n//2

for k in range(s):

    a=0

    for i in range(k+1,n-k-1):

        a+=matrix[k][i]+matrix[n-k-1][i]+matrix[i][k]+matrix[i][n-k-1]

    a+=matrix[k][k]+matrix[k][n-1-k]+matrix[n-1-k][k]+matrix[n-1-k][n-1-k]

    sumn.append(a)

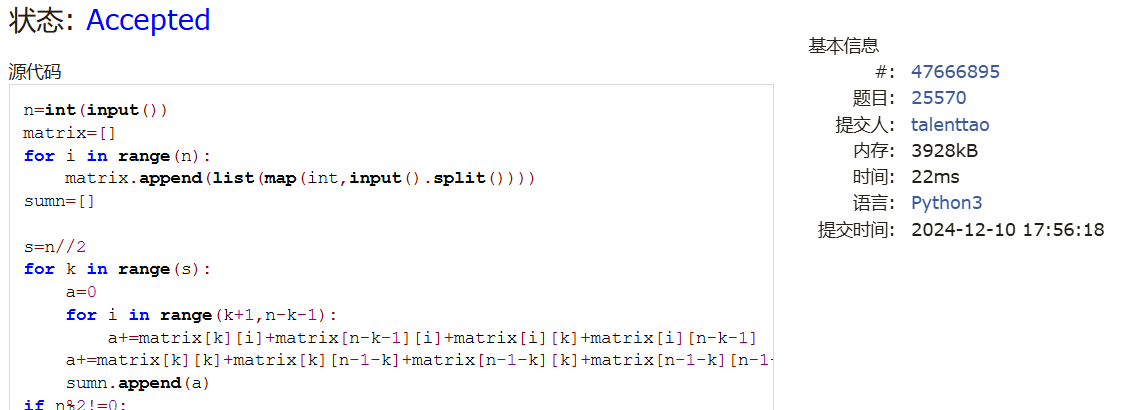
if n%2!=0:

    sumn.append(matrix[s][s])

print(max(sumn))

```

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：

一开始用dp一直超时，后来发现可以用最小堆。

代码：

```python

import heapq

n=int(input())

lst=list(map(int,input().split()))

a=[]

hp=0

num=0

for i in lst:

    if hp+i>=0:

        hp+=i

        num+=1

        if i<0:

            heapq.heappush(a,i)

    elif a and i>a[0]:

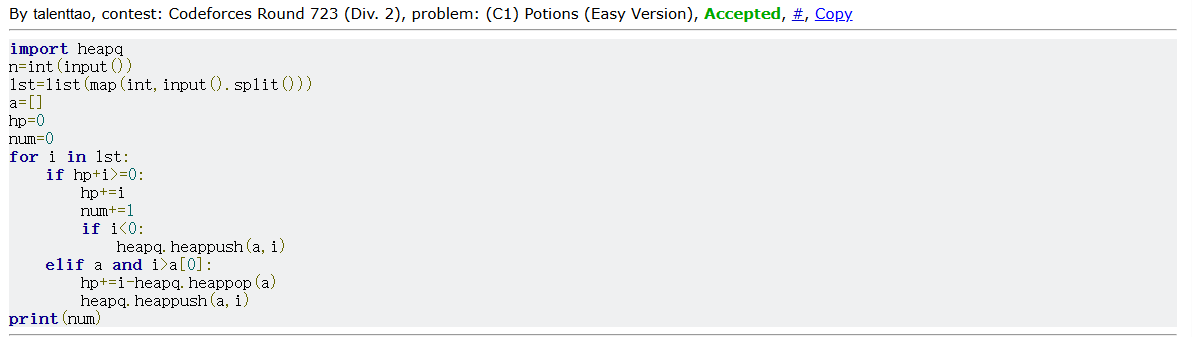
        hp+=i-heapq.heappop(a)

        heapq.heappush(a,i)

print(num)

```

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：

使用两个栈来实现功能

代码：

```python

a = []

b = []

while True:

    try:

        s = input().split()

        if s[0] == "min":

            if b:

                print(b[-1])

        elif s[0] == "pop":

            if a:

                a.pop()

                if b:

                    b.pop()

        else:

            weight = int(s[1])

            a.append(weight)

            if not b:

                b.append(weight)

            else:

                k = b[-1]

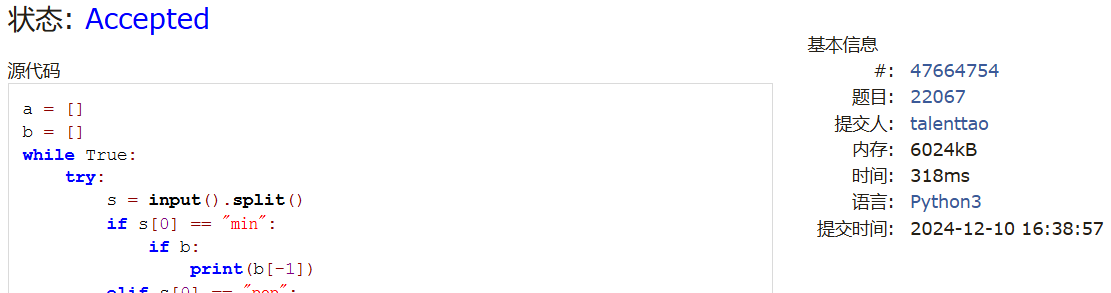
                b.append(min(k, weight))

    except EOFError:

        break

```

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：

一开始没有把消耗的体力作为第一个元素塞进堆里，导致浪费了较多时间

代码：

```python

import heapq

def dijkstra(start,end,hill):

    consume=[[float('inf')]\*n for \_ in range(m)]

    x0,y0=start

    xe,ye=end

    if hill[x0][y0]=='#' or hill[xe][ye]=='#':

        return 'NO'

    consume[x0][y0]=0

    pq=[(0,x0,y0)]

    while pq:

        con,x,y=heapq.heappop(pq)

        if (x,y)==end:

            return con

        for i in range(4):

            x1=x+dx[i]

            y1=y+dy[i]

            if 0<=x1<m and 0<=y1<n and hill[x1][y1]!='#':

                con1=con+abs(int(hill[x][y])-int(hill[x1][y1]))

                if con1<consume[x1][y1]:

                    consume[x1][y1]=con1

                    heapq.heappush(pq,(con1,x1,y1))

    return 'NO'

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

hill=[]

for i in range(m):

    hill.append(list(map(str,input().split())))

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

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

out=[]

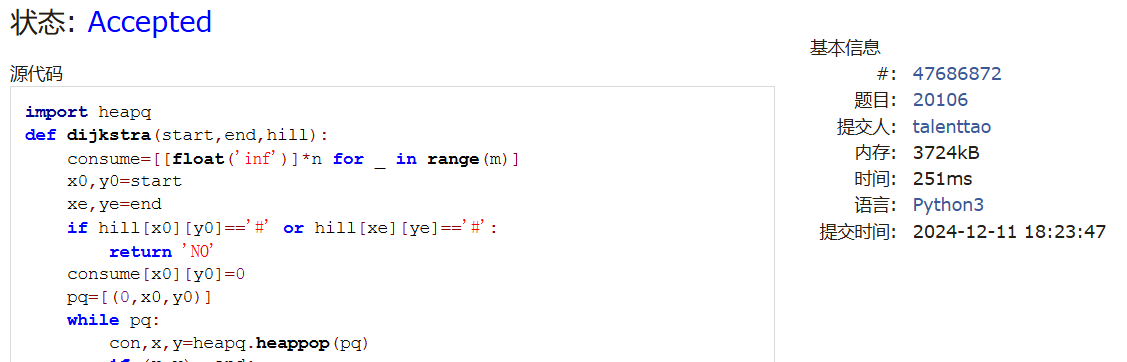
for i in range(p):

    a,b,c,d=map(int,input().split())

    print(dijkstra((a,b),(c,d),hill))

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



**### 04129: 变换的迷宫**

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

思路：

Bfs即可

代码：

```python

from collections import deque

def find(maze,r,c):

    for i in range(r):

        for j in range(c):

            if maze[i][j]=='S':

                return (i,j)

def bfs(maze,r,c,k):

    x0,y0=find(maze,r,c)

    a=deque()

    a.append((x0,y0,0,0))

    visited=set()

    visited.add((x0,y0,0))

    while a:

        x,y,mod,step=a.popleft()

        if maze[x][y]=='E':

            return step

        for i in range(4):

            x1=x+dx[i]

            y1=y+dy[i]

            if 0<=x1<r and 0<=y1<c:

                nmod=(mod+1)%k

                if maze[x1][y1]=='#' and nmod!=0:

                        continue

                if (x1,y1,nmod) not in visited:

                    a.append((x1,y1,nmod,step+1))

                    visited.add((x1,y1,nmod))

    return 'Oop!'

t=int(input())

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

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

for \_ in range(t):

    r,c,k=map(int,input().split())

    maze=[]

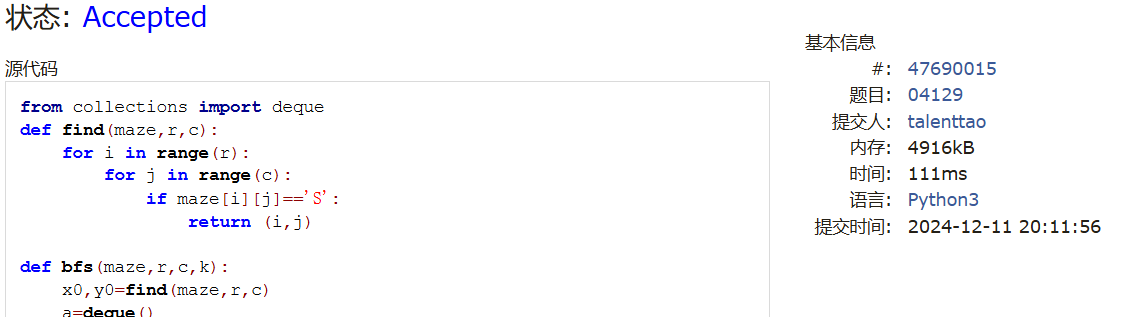
    for i in range(r):

        maze.append(input())

    print(bfs(maze,r,c,k))

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



**## 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>

本周在持续跟进每日练习，感觉作业相对之前做起来比较轻松，只是有关“堆”的题目由于之前没碰到过因此第一遍没写出来。期末将至，要每日练习以保持手感。