Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by 同学的姓名、院系

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：

借用了提示，只要找到在第一次出现[a/b] >= 2的时候谁是先手就可以

代码：

```python
def dfs(count, a, b):
    max_n, min_n = max(a, b), min(a, b)
    if max_n >= 2*min_n or max_n == min_n:
        return count
    else:
        return dfs(count + 1, max_n - min_n, min_n)

while True:
    a, b = map(int, input().split())
    if (a, b) == (0, 0):
        break
    print('win' if dfs(0, a, b)%2 == 0 else 'lose')
```

代码运行截图 （至少包含有"Accepted"）

代码提交状态： Accepted

25570: 洋葱

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

思路：

先把实心的每一层求出来，减去上一层实心的即可.

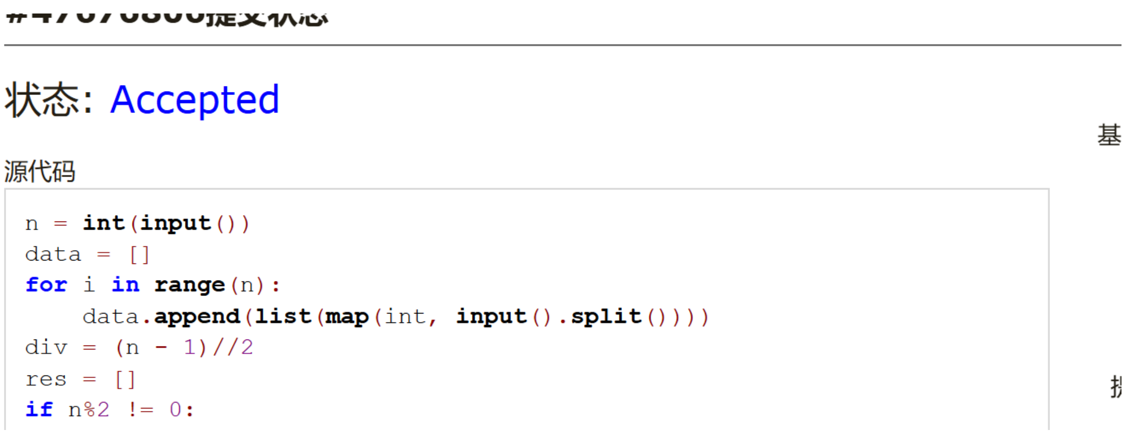
代码：

```
n = int(input())
data = []
for i in range(n):
    data.append(list(map(int, input().split())))
div = (n - 1)//2
res = []
if n%2 != 0:
    res.append(data[div][div])
else:
    res.append(sum(data[div][div:div + 2]) + sum(data[div + 1][div:div + 2]))

for i in range(div - 1, -1, -1):
    add = 0
    for j in range(i, n - i):
        add += sum(data[j][i: n - i])
    res.append(add - sum(res))

print(max(res))
```

代码运行截图 ==（至少包含有"Accepted"）==



#47670800提交状态

状态: Accepted

基

源代码

```
n = int(input())
data = []
for i in range(n):
    data.append(list(map(int, input().split())))
div = (n - 1)//2
res = []
if n%2 != 0:
```

1526C1. Potions(Easy Version)

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

思路：

很有趣的题，一开始没想到要用headq，后来看了一眼恍然大悟，加到笔记里了

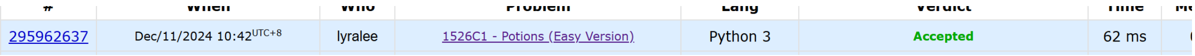
代码：

```python
import heapq

def max(n, potions):
    health = 0
    consumed = []
    for ele in potions:
        health += ele
        heapq.heappush(consumed, ele)
        if health < 0 and consumed:
            health -= consumed[0]
            heapq.heappop(consumed)
    return len(consumed)

n = int(input())
potions = list(map(int, input().split()))
print(max(n, potions))
```

代码运行截图 （至少包含有"Accepted"）



#	When	Who	Problem	Lang	Verdict	Time	Me
295962637	Dec/11/2024 10:42^{UTC+8}	lyralee	1526C1 - Potions (Easy Version)	Python 3	Accepted	62 ms	

22067: 快速堆猪

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

思路：

第一次学到辅助栈相关，借助题来看真的挺有意思的

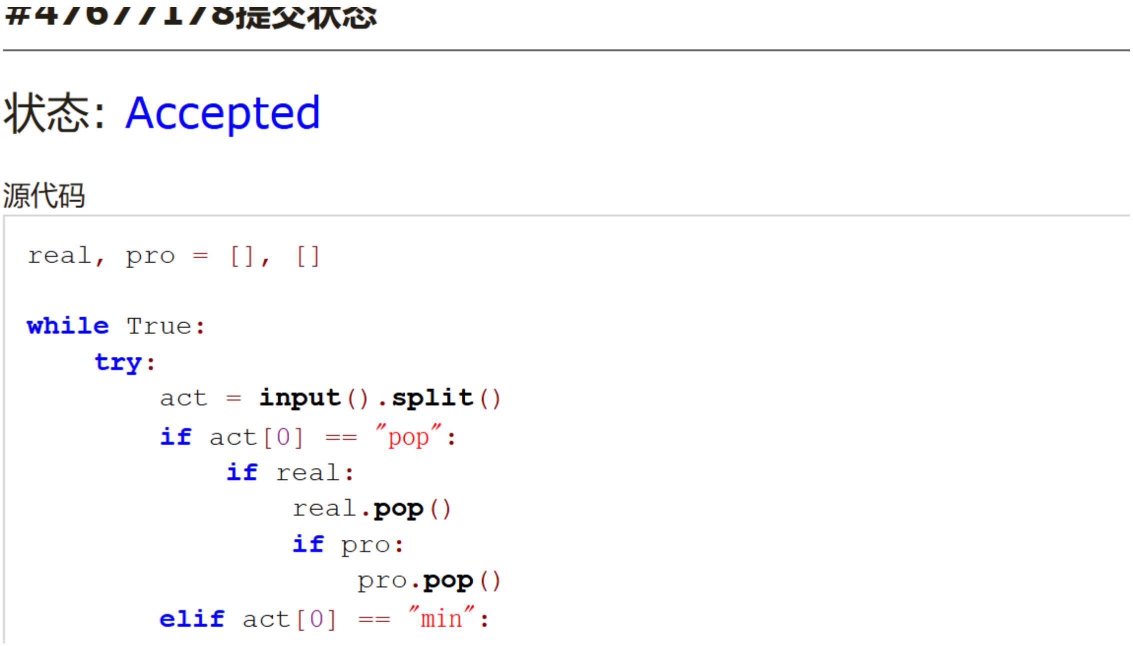
一个栈用来存储实际数据，一个用来存储当下最小的猪的数据

代码：

```python
real, pro = [], []

while True:
    try:
        act = input().split()
        if act[0] == "pop":
            if real:
                real.pop()
                if pro:
                    pro.pop()
        elif act[0] == "min":
            if pro:
                print(pro[-1])
        else:
            num = int(act[1])
            real.append(num)
            if not pro or pro[-1] >= num:
                pro.append(num)
            else:
                pro.append(pro[-1])
    except EOFError:
        break
```

代码运行截图 （至少包含有"Accepted"）



#47677178提交状态

状态: Accepted

源代码

```
real, pro = [], []

while True:
    try:
        act = input().split()
        if act[0] == "pop":
            if real:
                real.pop()
                if pro:
                    pro.pop()
        elif act[0] == "min":
```

20106: 走山路

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

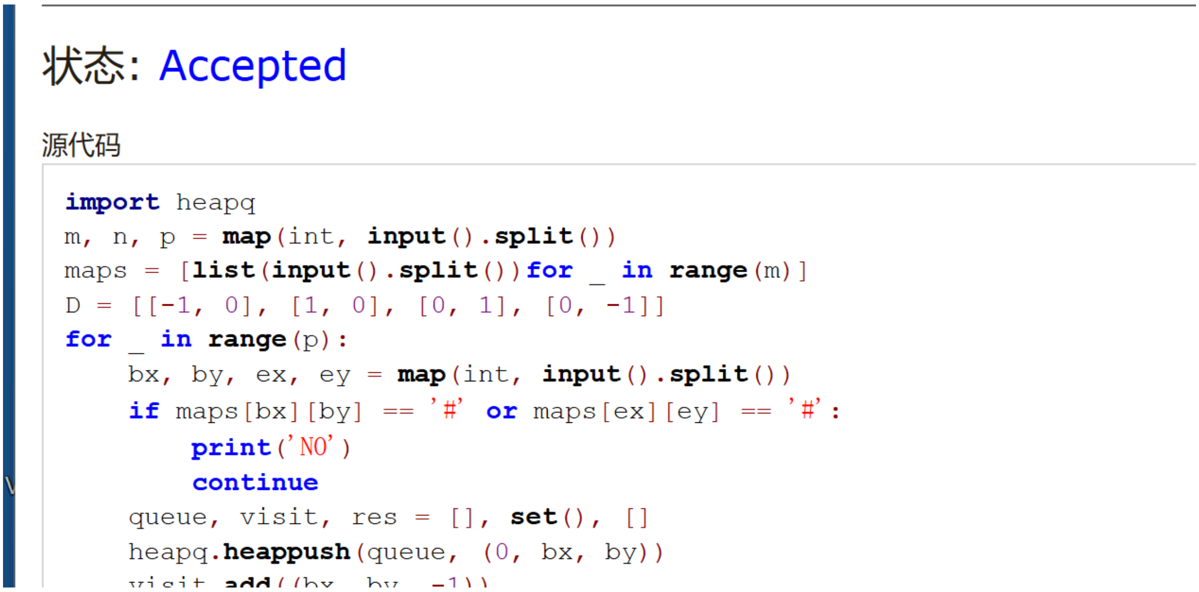
思路：

比较标准的题目，难点在于我还是不太熟悉整个算法流程，不看题解折腾了一个小时总算搞定了

代码：

```
import heapq
m, n, p = map(int, input().split())
maps = [list(input().split())for _ in range(m)]
D = [[-1, 0], [1, 0], [0, 1], [0, -1]]
for _ in range(p):
    bx, by, ex, ey = map(int, input().split())
    if maps[bx][by] == '#' or maps[ex][ey] == '#':
        print('NO')
        continue
    queue, visit, res = [], set(), []
    heapq.heappush(queue, (0, bx, by))
    visit.add((bx, by, -1))
    while queue:
        steps, x, y = heapq.heappop(queue)
        if (x, y) == (ex, ey):
            res.append(steps)
        for i in range(4):
            nx, ny = x + D[i][0], y + D[i][1]
            if 0 <= nx < m and 0 <= ny < n and maps[nx][ny] != '#' and (nx, ny,
i) not in visit:
                energy = steps + abs(int(maps[nx][ny]) - int(maps[x][y]))
                heapq.heappush(queue, (energy, nx, ny))
                visit.add((nx, ny, i))
    print(min(res) if res else 'NO')
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
import heapq
m, n, p = map(int, input().split())
maps = [list(input().split())for _ in range(m)]
D = [[-1, 0], [1, 0], [0, 1], [0, -1]]
for _ in range(p):
    bx, by, ex, ey = map(int, input().split())
    if maps[bx][by] == ’#’ or maps[ex][ey] == ’#’:
        print(’NO’)
        continue
    queue, visit, res = [], set(), []
    heapq.heappush(queue, (0, bx, by))
    visit.add((bx, by, -1))
```

04129: 变换的迷宫

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

思路:

（看提示学会）把入列条件设置为上一个周期（k时间）内有没有访问过这个点

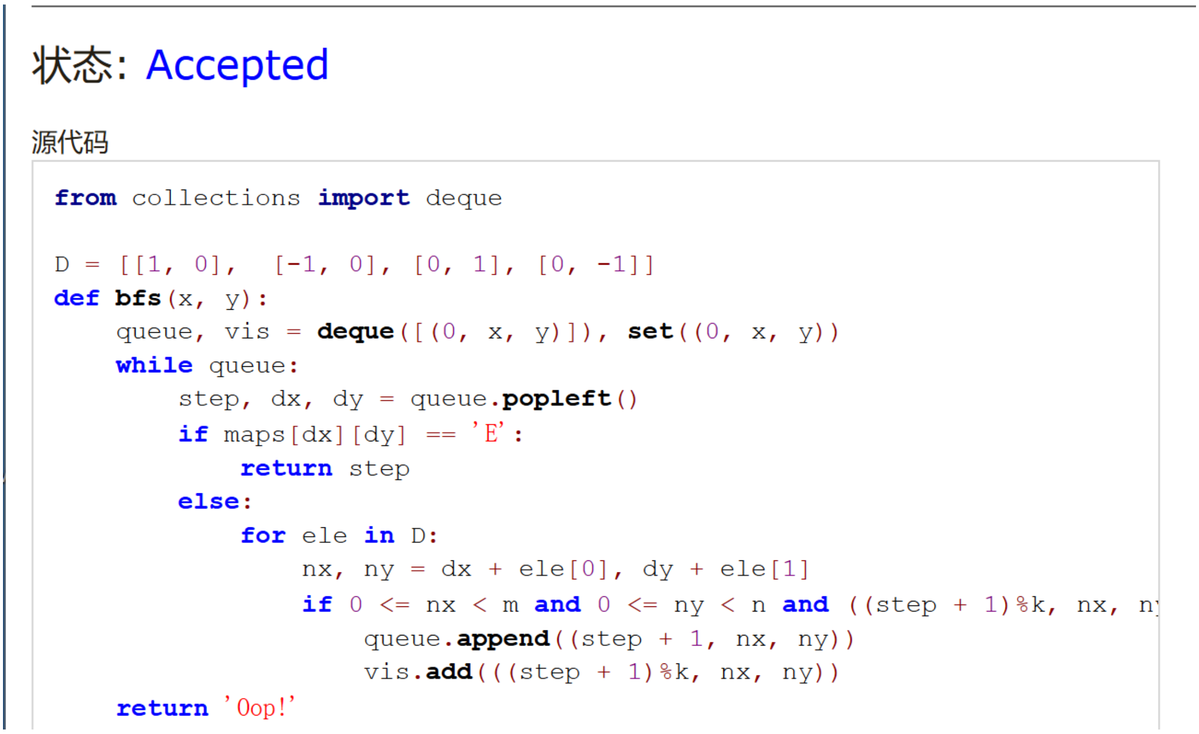
（折腾了半天终于自主搞定*或者说默写）bfs完整流程

代码:

```
from collections import deque

D = [[1, 0],  [-1, 0], [0, 1], [0, -1]]
def bfs(x, y):
    queue, vis = deque([(0, x, y)]), set((0, x, y))
    while queue:
        step, dx, dy = queue.popleft()
        if maps[dx][dy] == 'E':
            return step
        else:
            for ele in D:
                nx, ny = dx + ele[0], dy + ele[1]
                if 0 <= nx < m and 0 <= ny < n and ((step + 1)%k, nx, ny) not in vis and (maps[nx][ny] != '#' or (step + 1)% k == 0):
                    queue.append((step + 1, nx, ny))
                    vis.add(((step + 1)%k, nx, ny))
    return 'Oop!'
for _ in range(int(input())):
    maps = []
    m, n, k = map(int, input().split())
    for i in range(m):
        maps.append(list(input()))
    for i in range(m):
        for j in range(n):
            if maps[i][j] == 'S':
                print(bfs(i, j))
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
from collections import deque

D = [[1, 0],  [-1, 0], [0, 1], [0, -1]]
def bfs(x, y):
    queue, vis = deque([(0, x, y)]), set((0, x, y))
    while queue:
        step, dx, dy = queue.popleft()
        if maps[dx][dy] == ’E’:
            return step
        else:
            for ele in D:
                nx, ny = dx + ele[0], dy + ele[1]
                if 0 <= nx < m and 0 <= ny < n and ((step + 1)%k, nx, n
                    queue.append((step + 1, nx, ny))
                    vis.add(((step + 1)%k, nx, ny))
    return ’Oop!’
```

2. 学习总结和收获

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。

本周通过作业验证了初步具备不看答案完成bfs和dijkstra的实力，但是具体什么情况用哪一个算法还没搞清楚，在没有标签提示的情况下会有点吃亏，打算问一下ai

作业标题太妙了，题目内容“五味杂陈”，我的心情也五味杂陈

有点期末考试焦虑，希望最后结果能好吧