Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by 陈冠宇 工学院

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：其实并没有搞懂原理，但是写了个递归帮我硬算，好像对了（

代码：

```
def mygo(a,b):
    if b!=0 and int(a/b)>=2:      #提示给的信息，告诉我这种情况先手一定赢
        return True
    elif b==0:                    #一堆已经空了，说明此时的先手已经输了
        return False
    else:                         #取一轮之后互换先后手
        return not mygo(b,a-b)

while True:
    a,b=map(int,input().split())
    if a==0 and b==0:
        break
    else:
        if a<b:
            a,b=b,a
        if mygo(a,b):
            print("win")
        else:
            print("lose")
```

代码运行截图 （至少包含有"Accepted"）

```python
def mygo(a,b):
    if b!=0 and int(a/b)>=2:
        return True
    elif b==0:
        return False
    else:
        return not mygo(b,a-b)

while True:
    a,b=map(int,input().split())
    if a==0 and b==0:
        break
    else:
        if a<b:
            a,b=b,a
        if mygo(a,b):
            print("win")
        else:
            print("lose")
```

数据有点弱吗？可以申请加强数据

调试代码 提交答案

代码提交状态： Accepted

25570: 洋葱

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

思路：

代码：

```python
def max_layer_sum(matrix, n):
    max_sum = 0  # 记录最大层和
    num_layers = (n + 1) // 2  # 矩阵的总层数

    for layer in range(num_layers):
        layer_sum = 0

        # 遍历顶部行
        for col in range(layer, n - layer):
            layer_sum += matrix[layer][col]

        # 遍历底部行
        if n - layer - 1 > layer:  # 避免单行重复
            for col in range(layer, n - layer):
                layer_sum += matrix[n - layer - 1][col]

        # 遍历左侧列
        for row in range(layer + 1, n - layer - 1):  # 避免单列重复
            layer_sum += matrix[row][layer]

        # 遍历右侧列
        if n - layer - 1 > layer:  # 避免单列重复
            for row in range(layer + 1, n - layer - 1):
                layer_sum += matrix[row][n - layer - 1]

        # 更新最大层和
        max_sum = max(max_sum, layer_sum)

    return max_sum
```

```
# 输入处理
n = int(input())
matrix = [list(map(int, input().split())) for _ in range(n)]

# 输出结果
print(max_layer_sum(matrix, n))
```

代码运行截图 ==（至少包含有"Accepted"）==

#47748211提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def max_layer_sum(matrix, n):
    max_sum = 0  # 记录最大层和
    num_layers = (n + 1) // 2  # 矩阵的总层数

    for layer in range(num_layers):
        layer_sum = 0

        # 遍历顶部行
        for col in range(layer, n - layer):
            layer_sum += matrix[layer][col]

        # 遍历底部行
        if n - layer - 1 > layer:  # 避免单行重复
            for col in range(layer, n - layer):
```

基本信息

#: 47748211
题目: 25570
提交人: 陈冠宇(24n2400011004)
内存: 3996kB
时间: 23ms
语言: Python3
提交时间: 2024-12-15 13:34:34

1526C1. Potions(Easy Version)

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

思路：三维数组dp【前i瓶能喝的最多数量】【此时剩余生命值】【标记此时哪些已经喝过】，如果遇到值就喝，遇到负值就比较其与前面已喝过各瓶中“最毒的”一瓶，若将“最毒的”吐出来之后就能喝下当下这瓶并且剩余生命值增加了，就喝当下这瓶；否则不喝

代码：

```
def mygo(i,num_list,dp):
    dp[i][1]=dp[i-1][1]
    md=0
    for j in range(0,i-1):
        if num_list[j]<0 and md>num_list[j] and dp[i-1][2][j]:
            jj=j
            md=num_list[j]
    drink=dp[i-1][1]+num_list[i-1]-md
    not_drink=dp[i-1][1]
    if drink>not_drink:
        dp[i][1]=drink
        dp[i][2][i-1]=True
        dp[i][2][jj]=False
    dp[i][0]=dp[i-1][0]

n=int(input())
num_list=list(map(int,input().split()))
```

```
drunk=[False]*n
dp=[[0,0,drunk] for i in range(n+1)]
#print(dp)
for i in range(1,n+1):
    if num_list[i-1]>=0 or dp[i-1][1]+num_list[i-1]>=0:
        dp[i]=[dp[i-1][0]+1,dp[i-1][1]+num_list[i-1],dp[i-1][2]]
        dp[i][2][i-1]=True
    else:
        mygo(i,num_list,dp)

print(dp[-1][0])
```

代码运行截图 （至少包含有"Accepted"）

By chriscgy, contest: Codeforces Round 723 (Div. 2), problem: (C1) Potions (Easy Version), **Accepted**, #, Copy

```
def mygo(i,num_list,dp):
    dp[i][1]=dp[i-1][1]
    md=0
    for j in range(0,i-1):
        if num_list[j]<0 and md>num_list[j] and dp[i-1][2][j]:
            jj=j
            md=num_list[j]
    drink=dp[i-1][1]+num_list[i-1]-md
    not_drink=dp[i-1][1]
    if drink>not_drink:
        dp[i][1]=drink
        dp[i][2][i-1]=True
        dp[i][2][jj]=False
    dp[i][0]=dp[i-1][0]

n=int(input())
num_list=list(map(int,input().split()))


drunk=[False]*n
dp=[[0,0,drunk] for i in range(n+1)]
#print(dp)
for i in range(1,n+1):
    if num_list[i-1]>=0 or dp[i-1][1]+num_list[i-1]>=0:
        dp[i]=[dp[i-1][0]+1,dp[i-1][1]+num_list[i-1],dp[i-1][2]]
        dp[i][2][i-1]=True
    else:
        mygo(i,num_list,dp)

print(dp[-1][0])
```

22067: 快速堆猪

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

思路：一时没想到可以构建最轻猪列表

代码：

```
def mygo(word):
    if len(word)==1:
        if word[0]=='pop':
```

```
            return -1
        if word[0]=='min':
            return -2
    return int(word[1])

light=[]
pigs=[]
while True:
    try:
        word=input().split()
    except EOFError:
        break
    x=mygo(word)
    if x==-1:
        try:
            pigs.pop()
            light.pop()
        except IndexError:
            continue

    elif x==-2:
        try:
            print(light[-1])
        except IndexError:
            continue

    elif x>=0:
        pigs.append(x)
        if len(light)==0:
            light.append(x)
        else:
            light.append(min(x,light[-1]))
    #print(pigs)
```

代码运行截图（至少包含有"Accepted"）

#47763715提交状态 查看 提交 统计 提问

状态: Accepted

基本信息
#: 47763715
题目: 22067
提交人: 陈冠宇(24n2400011004)
内存: 6820kB
时间: 336ms
语言: Python3
提交时间: 2024-12-16 10:24:11

源代码

```
def mygo(word):
    if len(word)==1:
        if word[0]=='pop':
            return -1
        if word[0]=='min':
            return -2
    return int(word[1])

light=[]
pigs=[]
while True:
    try:
        word=input().split()
    except EOFError:
        break
    x=mygo(word)
    if x==-1:
        try:
            pigs.pop()
            light.pop()
        except IndexError:
            continue

    elif x==-2:
        try:
            print(light[-1])
        except IndexError:
            continue

    elif x>=0:
        pigs.append(x)
        if len(light)==0:
            light.append(x)
        else:
            light.append(min(x,light[-1]))
    #print(pigs)
```

20106: 走山路

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

思路:

代码:

```
import heapq
def dijkstra(m,n,chizu,start,end):
    directions=[(1,0),(0,1),(-1,0),(0,-1)]
    dist=[[float('inf') for _ in range(n)] for _ in range(m)]
    if chizu[start[0]][start[1]]=='#' or chizu[end[0]][end[1]]=='#':
        return"NO"
    pq=[]
    dist[start[0]][start[1]]=0
    heapq.heappush(pq,(dist[start[0]][start[1]],start[0],start[1]))
    while pq:
        cost,x,y=heapq.heappop(pq)
        if (x,y) ==end:
            return cost
        if cost>dist[x][y]:
            continue
        for dx,dy in directions:
            nx,ny=x+dx,y+dy
            if 0<=nx<m and 0<=ny<n and chizu[nx][ny]!='#':
```

```
                new_cost=cost+abs(int(chizu[nx][ny])-int(chizu[x][y]))
                if new_cost<dist[nx][ny]:
                    dist[nx][ny]=new_cost
                    heapq.heappush(pq,(dist[nx][ny],nx,ny))
    return 'NO'


m,n,p=map(int,input().split())
chizu=[]
for i in range(m):
    chizu.append(input().split())
for i in range(p):
    start_x, start_y, end_x, end_y = map(int, input().split())
    start = (start_x, start_y)
    end = (end_x, end_y)
    print(dijkstra(m,n,chizu,start,end))
```

代码运行截图 （至少包含有"Accepted"）

#47764361提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq
def dijkstra(m,n,chizu,start,end):
    directions=[(1,0),(0,1),(-1,0),(0,-1)]
    dist=[[float('inf') for _ in range(n)] for _ in range(m)]
    if chizu[start[0]][start[1]]=='#' or chizu[end[0]][end[1]]=='#':
        return"NO"
    pq=[]
    dist[start[0]][start[1]]=0
    heapq.heappush(pq,(dist[start[0]][start[1]],start[0],start[1]))
    while pq:
        cost,x,y=heapq.heappop(pq)
        if (x,y) ==end:
            return cost
        if cost>dist[x][y]:
            continue
        for dx,dy in directions:
            nx,ny=x+dx,y+dy
            if 0<=nx<m and 0<=ny<n and chizu[nx][ny]!='#':
                new_cost=cost+abs(int(chizu[nx][ny])-int(chizu[x][y]))
                if new_cost<dist[nx][ny]:
                    dist[nx][ny]=new_cost
                    heapq.heappush(pq,(dist[nx][ny],nx,ny))
    return 'NO'


m,n,p=map(int,input().split())
chizu=[]
for i in range(m):
    chizu.append(input().split())
for i in range(p):
    start_x, start_y, end_x, end_y = map(int, input().split())
    start = (start_x, start_y)
    end = (end_x, end_y)
    print(dijkstra(m,n,chizu,start,end))
```

基本信息

#:	47764361
题目:	20106
提交人:	陈冠宇(24n2400011004)
内存:	3740kB
时间:	227ms
语言:	Python3
提交时间:	2024-12-16 11:17:11

04129: 变换的迷宫

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

思路：之前每日选做做了这题

最重要的地方在于visited数组要有一个计数器，如果每次走到某一点时步数和前一次走到相差k的整数倍，则说明该点相当于已经访问过了

代码：

```
from collections import deque

def bfs(x,y,r,c,ari_chizu,nashi_chizu,k):
    dd=deque([(start_x,start_y,0)])
    visited={(0,start_x,start_y)}
    current_chizu=ari_chizu
    dirs = [(1, 0), (-1, 0), (0, 1), (0, -1)]
    while dd:
        x,y,counter=dd.popleft()
        temp=(counter+1)%k

        if temp==0:
            current_chizu=nashi_chizu
        else:
            current_chizu=ari_chizu

        for dx,dy in dirs:
            nx,ny=x+dx,y+dy
            if 0<=nx<r and 0<=ny<c and (temp,nx,ny) not in visited:
                if current_chizu[nx][ny]==2:
                    return counter+1
                elif current_chizu[nx][ny]==0:
                    visited.add((temp,nx,ny))
                    dd.append((nx,ny,counter+1))

    return -1

T=int(input())
for tt in range(T):
    r,c,k=map(int,input().split())
    ari_chizu=[[0 for _ in range(c)]for _ in range(r)]
    nashi_chizu = [[0 for _ in range(c)] for _ in range(r)]
    for i in range(r):
        x=input()
        for j in range(len(x)):
            if x[j]=='.':
                ari_chizu[i][j]=0
                nashi_chizu[i][j]=0
            elif x[j]=='#':
                ari_chizu[i][j]=1
                nashi_chizu[i][j]=0
            elif x[j]=='S':
                ari_chizu[i][j]=0
                nashi_chizu[i][j]=0
                start_x=i
```

```
                start_y=j
            elif x[j]=='E':
                ari_chizu[i][j]=2
                nashi_chizu[i][j]=2

    ans= bfs(start_x,start_y,r,c,ari_chizu,nashi_chizu,k)
    if ans==-1:
        print('Oop!')
    else:
        print(ans)
```

代码运行截图 （至少包含有"Accepted"）

#47536467提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque

def bfs(x,y,r,c,ari_chizu,nashi_chizu,k):
    dd=deque([(start_x,start_y,0)])
    visited={(0,start_x,start_y)}
    current_chizu=ari_chizu
    dirs = [(1, 0), (-1, 0), (0, 1), (0, -1)]
    while dd:
        x,y,counter=dd.popleft()
        temp=(counter+1)%k

        if temp==0:
            current_chizu=nashi_chizu
        else:
            current_chizu=ari_chizu

        for dx,dy in dirs:
            nx,ny=x+dx,y+dy
            if 0<=nx<r and 0<=ny<c and (temp,nx,ny) not in visited:
                if current_chizu[nx][ny]==2:
                    return counter+1
                elif current_chizu[nx][ny]==0:
                    visited.add((temp,nx,ny))
                    dd.append((nx,ny,counter+1))

    return -1

T=int(input())
for tt in range(T):
    r,c,k=map(int,input().split())
    ari_chizu=[[0 for _ in range(c)]for _ in range(r)]
    nashi_chizu = [[0 for _ in range(c)] for _ in range(r)]
    for i in range(r):
        x=input()
        for j in range(len(x)):
            if x[j]=='.':
                ari_chizu[i][j]=0
                nashi_chizu[i][j]=0
            elif x[j]=='#':
                ari_chizu[i][j]=1
                nashi_chizu[i][j]=0
            elif x[j]=='S':
                ari_chizu[i][j]=0
                nashi_chizu[i][j]=0
                start_x=i
                start_y=j
            elif x[j]=='E':
                ari_chizu[i][j]=2
                nashi_chizu[i][j]=2

    ans= bfs(start_x,start_y,r,c,ari_chizu,nashi_chizu,k)
    if ans==-1:
        print('Oop!')
    else:
        print(ans)
```

基本信息

#:	47536467
题目:	04129
提交人:	陈冠宇(24n2400011004)
内存:	5220kB
时间:	121ms
语言:	Python3
提交时间:	2024-12-03 17:14:31

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。

机考临近，有点慌。。感觉自己debug能力还有待提升，容易在细节上耗费大量时间