Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by <mark>同学的姓名、院系</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：题目已经给出了思路，即找到第一次 a//b>=2 时谁是先手就可以。（1h）

代码：

```
memory = {}
def stone_game(a, b):
    if (a, b) in memory:
        return memory[(a, b)]
    if a == 0 or b == 0:
        return a > b
    flag = False
    large = max(a, b)
    small = min(a, b)
```

```
    k = 1
    while k * small <= large:
        new_large = large - k * small
        new_small = small
        result = not stone_game(new_large, new_small)
        if result:
            flag = True
            break
        k += 1
    memory[(a, b)] = flag
    return flag
while True:
    a, b = map(int, input().split())
    if a == 0 and b == 0:
        break
    print("win" if stone_game(a, b) else "lose")
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

挑战模式

```python
memory = {}
def stone_game(a, b):
    if (a, b) in memory:
        return memory[(a, b)]
    if a == 0 or b == 0:
        return a > b
    flag = False
    large = max(a, b)
    small = min(a, b)
    k = 1
    while k * small <= large:
        new_large = large - k * small
        new_small = small
        result = not stone_game(new_large, new_small)
        if result:
            flag = True
            break
        k += 1
    memory[(a, b)] = flag
    return flag
while True:
    a, b = map(int, input().split())
    if a == 0 and b == 0:
        break
    print("win" if stone_game(a, b) else "lose")
```

数据有点弱吗？可以申请加强数据

25570: 洋葱

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

思路：如果为空矩阵直接返回 0，如果只有一个元素输出该值即可，然后遍历边缘的元素，确定方向，再对第二层进行递归，注意长度变成 n-2.

最后输出即可（2h）

代码：

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

代码运行截图 ==（至少包含有"Accepted"）==

#47735962提交状态 查看

状态: Accepted

源代码

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

基本信息

#: 477
题目: 255
提交人: 24n
内存: 393
时间: 23n
语言: Pytl
提交时间: 202

1526C1. Potions(Easy Version)

greedy, dp, data structures, brute force, *1500,

https://codeforces.com/problemset/problem/1526/C1

思路:从零开始学习堆的概念,感觉作用大致是找出‘最差’的那一个,并用 dp 表示喝下第 i 瓶药水后的生命值,如果生命值大于 0 就继续,如果小于

0 就从堆中找出‘最差的一个’减掉并继续遍历。（3h）

代码：

```
import heapq

n = int(input())

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

dp = [0] *(n+1)

ans = 0

q = []

for i in range(1,n+1):

    heapq. heappush(q, s[i-1])

    dp[i] = dp[i - 1] + s[i - 1]

    if dp[i]>=0:

        ans += 1

    else:

        worst_potion = heapq. heappop(q)

        dp[i]-=worst_potion

print(ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

Who	Problem	Lang	Verdict
EuphoriaJ	1526C1 - Potions (Easy Version)	Python 3	Accepted

22067: 快速堆猪

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

思路：该题运用了 heapq 与 defaultdict，对于 pop 命令，判断 s 是否为空，若不是取走最后一个即可，对于 min 命令，先找出堆里最小的元素

，再通过 deleted 的值判断是否实际在猪堆里，对于 Push 命令，添加至 stack 中即可。（4h）

代码：

```
import heapq
from collections import defaultdict
stack=[]
```

```
weight=[]
deleted=defaultdict(int)
while True:
    try:
        s=input().split()
        if s[0]=='pop':
            if stack:
                deleted[stack.pop()]+=1
        elif s[0]=='min':
            if stack:
                while True:
                    x=heapq.heappop(weight)
                    if not deleted[x]:
                        heapq.heappush(weight,x)
                        print(x)
                        break
                    deleted[x]-=1
        else:
            n=int(s[1])
            stack.append(n)
            heapq.heappush(weight,n)
    except EOFError:
```

```
        break
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

题目 排名 状态 提问

#47736925提交状态 查看

状态: Accepted

源代码

```
import heapq
from collections import defaultdict
stack=[]
weight=[]
deleted=defaultdict(int)
while True:
    try:
        s=input().split()
        if s[0]=='pop':
            if stack:
                deleted[stack.pop()]+=1
        elif s[0]=='min':
            if stack:
                while True:
                    x=heapq.heappop(weight)
                    if not deleted[x]:
                        heapq.heappush(weight,x)
                        print(x)
                        break
```

基本信息

#: 47736
题目: 22067
提交人: 24n24
内存: 6844k
时间: 347m
语言: Pytho
提交时间: 2024-

```
### 20106: 走山路
```

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

思路:基本上可以按照 dijkstra 的模版来写,感觉 dijkstra 与 bfs 的套路很像，但具体区别在哪里还是不太明白（耗时半天往上）

代码:

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

```
                new_cost=cost+abs(int(grid[nx][ny])-in
t(grid[x][y])) if grid[nx][ny]!='#' else float('inf')
                if new_cost<dist[nx][ny]:
                    dist[nx][ny]=new_cost
                    heapq.heappush(pq,(new_cost,nx,ny)
)
    return 'NO'
m,n,p=map(int,input().split())
grid=[]
for _ in range(m):
    grid.append(input().split())
for _ in range(p):
    sx,sy,ex,ey=map(int,input().split())
    result=min_cost_to_travel(m,n,grid,(sx,sy),(ex,ey)
)
    print(result)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47738531提交状态

查看 提交 统计

状态: Accepted

源代码

```
import heapq
directions=[(-1,0),(1,0),(0,-1),(0,1)]
def min_cost_to_travel(m,n,grid,start,end):
    dist=[[float('inf')]*n for _ in range(m)]
    if grid[start[0]][start[1]]=='#' or grid[end[0]][end[1]]=='#':
        return 'NO'
    pq=[]
    heapq.heappush(pq,(0,start[0],start[1]))
    dist[start[0]][start[1]]=0
    while pq:
        cost,x,y=heapq.heappop(pq)
        if (x,y)==end:
            return cost
        for dx,dy in directions:
            nx,ny=x+dx,y+dy
            if 0<=nx<m and 0<=ny<n and grid[nx][ny]!='#':
                new_cost=cost+abs(int(grid[nx][ny])-int(grid[x][y])) if
                if new_cost<dist[nx][ny]:
                    dist[nx][ny]=new_cost
                    heapq.heappush(pq,(new_cost,nx,ny))
    return 'NO'
m,n,p=map(int,input().split())
grid=[]
for _ in range(m):
    grid.append(input().split())
for _ in range(p):
```

基本信息

#: 47738531
题目: 20106
提交人: 24n2400011009
内存: 4012kB
时间: 235ms
语言: Python3
提交时间: 2024-12-14 19:11:

04129: 变换的迷宫

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

思路：摆烂了，不会一点，直接放弃，看的群里的同学的题解，使用了堆（通过 heapq 模块实现）来辅助进行类似贪心策略的路径搜索在循环中不断取出堆顶元素（代表当前代价最小的待探索位置），判断是否到达终点，如果到达则返回当前的路径代价。如果未到达

终点，就对当前位置向四个方向进行探索。（耗时 inf）

代码：

```
import heapq
from math import inf
directions = [(1,0),(0,1),(-1,0),(0,-1)]
def check(x,y,s):
    if M[x][y] == 1:
        return False
    if k == 2 and s != 0:
        return True
    for dx,dy in directions:
        if 0 <= (nx:=x+dx) < a and 0 <= (ny:=y+dy) < b and
M[nx][ny] == 0:
            return True
    return False
```

```
def best_way(points):
    while points:
        s,x,y = heapq.heappop(points)
        if x == ex and y == ey:
```

```
            return s
        for dx,dy in directions:
            if 0 <= (nx:=x+dx) < a and 0 <= (ny:=y+dy) <
b:
                if M[nx][ny] == 1 and check(x,y,s):
                    if S[nx][ny] > (ns:=(1+s//k)*k) and
s%2 != ns%2:
                        S[nx][ny] = ns
                        if not C[nx][ny]:
                            heapq.heappush(points,(ns,n
x,ny))
                            C[nx][ny] = True
                    elif S[nx][ny] > (ns:=ns+k) and
s%2 != ns%2:
                        S[nx][ny] = ns
                        if not C[nx][ny]:
                            heapq.heappush(points,(ns,n
x,ny))
                            C[nx][ny] = True
                elif M[nx][ny] == 0 and S[nx][ny] >
(ns:=s+1):
                    S[nx][ny] = ns
```

```
                    if not C[nx][ny]:
                        heapq.heappush(points,(ns,nx,ny)
)
                        C[nx][ny] = True
    return "Oop!"
Ans = []
T = int(input())
for t in range(T):
    a,b,k = map(int,input().split())
    M = [[0]*b for _ in range(a)]
    for i in range(a):
        l = input()
        for j in range(b):
            if l[j] == 'S':
                sx,sy = i,j
            elif l[j] == 'E':
                ex,ey = i,j
            elif l[j] == '#':
                M[i][j] = 1
```

```
    S = [[inf]*b for _ in range(a)]
    S[sx][sy] = 0
```

```
    C = [[False]*b for _ in range(a)]
    C[sx][sy] = True
    Ans.append(best_way([(0,sx,sy)]))
```

```
for ans in Ans:
    print(ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
import heapq
from math import inf
directions = [(1,0),(0,1),(-1,0),(0,-1)]
def check(x,y,s):
    if M[x][y] == 1:
        return False
    if k == 2 and s != 0:
        return True
    for dx,dy in directions:
        if 0 <= (nx:=x+dx) < a and 0 <= (ny:=y+dy) < b and M[nx][ny] ==
            return True
    return False

def best_way(points):
    while points:
        s,x,y = heapq.heappop(points)
        if x == ex and y == ey:
            return s
        for dx,dy in directions:
            if 0 <= (nx:=x+dx) < a and 0 <= (ny:=y+dy) < b:
                if M[nx][ny] == 1 and check(x,y,s):
                    if S[nx][ny] > (ns:=(1+s//k)*k) and s%2 != ns%2:
                        S[nx][ny] = ns
                        if not C[nx][ny]:
                            heapq.heappush(points,(ns,nx,ny))
```

基本信息

#: 47737545
题目: 04129
提交人: 24n2400011
内存: 3912kB
时间: 94ms
语言: Python3
提交时间: 2024-12-14

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>

也没觉得这周作业简单啊....还有两周机考，只能希望搜索碰见模版题，dp听天由命，希望两个简单题难度可以降一点，真的

有点担心要是一个题都做不出来该怎么办，这周刷了小几十道leetcode和晴问上的题，突然发现晴问有些简单题就开始用搜索，真的

打击人的心态...要是两个E题像晴问上的入门难度就好了（只是临死前的幻想）

快要被计概逼疯，花的时间比专业课还多还没有什么好效果，反正就是很让人绝望。。。