

Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Compiled by <mark>汤伟杰，信息管理系</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：

博弈题我觉得会用一个很不容易发现的最终结论，这个题的最终结论就在提示里面，我仍然觉得自己是想不到的，，然后代码的 dfs 的返回值是布尔值，在调用的时候可以 `return not dfs` 来模拟我方和对方是胜负，这个是从题解里面学到的。

代码：

```
def dfs(n,m):
    if n/m>=2 or n==m:
        return True
    return not dfs(m,n-m)

while True:
    n,m=map(int,input().split())
    if {n,m}=={0}:
        break
    if n<m:
        n,m=m,n
    if dfs(n,m):
        print('win')
    else:
        print('lose')
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

```

1 def dfs(n,m):
2     if n/m>=2 or n==m:
3         return True
4     return not dfs(m,n-m)
5
6 while True:
7     n,m=map(int,input().split())
8     if {n,m}=={0}:
9         break
10    if n<m:
11        n,m=m,n
12    if dfs(n,m):
13        print('win')
14    else:
15        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:

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:

```

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"）==

#47654870提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
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)
```

基本信息

#: 47654870
题目: 25570
提交人: 24n2400016635
内存: 3932kB
时间: 23ms
语言: Python3
提交时间: 2024-12-10 01:24:39

1526C1. Potions(Easy Version)

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

思路:

自己想了 dp, 没想出来转移方程, 群里代码也没看懂 (更主要的是自己想不出来), tutorial 里面的 dp 有点好理解, 但是 dp 并非记录喝下的数量而是健康数值, 最后找 dp[i][j] 的 j 的最大值, 有点新, 但是转移的思路中第 i 个喝还是不喝的想法有点熟悉。

问了 gpt 知道了贪心的策略, 遇到喝不下的药水, 那就看看以前喝下的有没有毒性更强的, 要是有的就“替换”掉, 这样既保证了喝的药水数量不变, 贪心了, 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。感觉这道题好难。

代码:

```
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>

By [twj_ink](#), contest: Codeforces Round 723 (Div. 2), problem: (C1) Potions (Easy Version), [Accepted](#), <#>, [Copy](#)

```

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)

```

22067: 快速堆猪

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

思路：

感觉题目的意图就是考栈和堆，第一次自己写没有用懒删除会超时，从题解里学到了懒删除（该删的时候先不在堆里面删因为没有这个方法，而用字典记录一下要删除的元素和对应的删除次数，等到要取最小值时再不断地取出堆顶元素，查看字典里面它的值是不是 0，如果不是说明它其实已经被删除了，那就扔掉继续取堆顶元素，用到 while 循环）。

同时也学到了辅助栈，辅助的作用是记录 $dp[i]$ ，表示到第 i 个元素为止，出现过的最小元素，疑似有点 dp 的味道。

代码：

```

import heapq
from collections import defaultdict
out=defaultdict(int)
stack,heap=[],[]
while True:
    try:
        s=input()
    except EOFError:
        break

```

```

if s=='pop' and stack:
    toss=stack.pop()
    out[toss]+=1
elif s=='min' and stack:
    while heap:
        curr_min=heapq.heappop(heap)
        if out[curr_min]==0:
            print(curr_min)
            heapq.heappush(heap,curr_min)
            break
        out[curr_min]-=1

elif s[-1].isdigit():
    n=int(s.split()[1])
    stack.append(n)
    heapq.heappush(heap,n)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47679380提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

import heapq
from collections import defaultdict
out=defaultdict(int)
stack,heap=[],[]
while True:
    try:
        s=input()
    except EOFError:
        break

    if s=='pop' and stack:
        toss=stack.pop()
        out[toss]+=1
    elif s=='min' and stack:
        while heap:
            curr_min=heapq.heappop(heap)
            if out[curr_min]==0:
                print(curr_min)
                heapq.heappush(heap,curr_min)
                break
            out[curr_min]-=1

    elif s[-1].isdigit():
        n=int(s.split()[1])
        stack.append(n)
        heapq.heappush(heap,n)

```

基本信息

#: 47679380
 题目: 22067
 提交人: 24n2400016635
 内存: 6100kB
 时间: 367ms
 语言: Python3
 提交时间: 2024-12-11 13:17:03

20106: 走山路

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

思路:

dijkstra 的思路: 把所有点的 dist (距离) 设为无穷。从起点开始找邻居, 如果加上边权后, 到邻居的距离小于 dist 中的记录, 就更新 dist 并加入优先队列。每次从优先队列中取出当前路径值最小的元素, 重复上述操作。

代码:

```
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>

状态: Accepted

源代码

```
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))
```

基本信息

#: 47694402
题目: 20106
提交人: 24n2400016635
内存: 3732kB
时间: 195ms
语言: Python3
提交时间: 2024-12-12 01:14:54

04129: 变换的迷宫

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

思路:

自己一直 WA，看了群里同学的思路的思考 1 和思考 2 后发现自己错误原因竟然完全吻合，一是要加时间，因为有的地方是可以在不同时间重复走的，状态不一样；二是当石头没有消失时，不能判断下一个位置是"."，而是要判断不是"#", 因为起点也是可以走的。考试遇到这题估计能气死，做不出来吧我也知道是 bfs，知道吧又找不到陷阱。。

代码:

```
from collections import deque
dx,dy=[0,-1,1,0],[-1,0,0,1]
def bfs(s,x,y,step):
    q=deque()
    q.append((x,y,step))
    inq=set()
    inq.add((x,y,step))
    while q:
        for _ in range(len(q)):
            x,y,step=q.popleft()
            for i in range(4):
```

```

        nx,ny=x+dx[i],y+dy[i]
        if 0<=nx<n and 0<=ny<m and (nx,ny,(step+1)%k) not in inq:
            if s[nx][ny]=='E':
                return step+1
            if ((step+1)%k!=0 and s[nx][ny]!='#') or ((step+1)%k==0):
                q.append((nx,ny,step+1))
                inq.add((nx,ny,(step+1)%k))

    return "Oops!"

for _ in range(int(input())):
    n,m,k=map(int,input().split())
    s=[input() for _ in range(n)]
    for i in range(n):
        for j in range(m):
            if s[i][j]=='S':
                print(bfs(s,i,j,0))

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```

from collections import deque
dx,dy=[0,-1,1,0],[-1,0,0,1]
def bfs(s,x,y,step):
    q=deque()
    q.append((x,y,step))
    inq=set()
    inq.add((x,y,step))
    while q:
        for _ in range(len(q)):
            x,y,step=q.popleft()
            for i in range(4):
                nx,ny=x+dx[i],y+dy[i]
                if 0<=nx<n and 0<=ny<m and (nx,ny,(step+1)%k) not in inq:
                    if s[nx][ny]=='E':
                        return step+1
                    if ((step+1)%k!=0 and s[nx][ny]!='#') or ((step+1)%k==0):
                        q.append((nx,ny,step+1))
                        inq.add((nx,ny,(step+1)%k))

    return "Oops!"

for _ in range(int(input())):
    n,m,k=map(int,input().split())
    s=[input() for _ in range(n)]
    for i in range(n):
        for j in range(m):
            if s[i][j]=='S':
                print(bfs(s,i,j,0))

```

基本信息

#: 47692355
 题目: 04129
 提交人: 24n2400016635
 内存: 4936kB
 时间: 134ms
 语言: Python3
 提交时间: 2024-12-11 21:49:11

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>

potion 和猪和山路都用到了 heapq，受益匪浅；取石子的博弈条件感觉想不到，但学到了 dfs 设置返回值为布尔值，从而可以 return not dfs()来模拟双方操作输赢；变换迷宫是 bfs 变形，细节好多，但是模板现在可以非常熟练地敲出来了（

先冲刺每日选做（因为可以学到好多），同时整理 **dp** 题的思路，希望期末能保持头脑清晰，把会做的题都拿下。