

Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Compiled by 陈俊逸、工院

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

02692: 假币问题

brute force, <http://cs101.openjudge.cn/practice/02692>

思路:

这个题老早就做过了，当时只会if else，但是这里我要推荐一下我的大佬室友（不是我们班的）的超强思路（我认为）：

开两个列表，记录每个硬币被怀疑的次数。如果even那么肯定是正常的，就+-一个很大的数就行了（这里我用的是114514），然后最后输出被怀疑次数最多的那个硬币并且看看最多的是在qing还是zhong列表里，对应着假币本身的轻重

代码:

```
def main():  
  
    l=  
{0: 'A', 1: 'B', 2: 'C', 3: 'D', 4: 'E', 5: 'F', 6: 'G', 7: 'H', 8: 'I', 9: 'J', 10: 'K', 11: 'L'}  
    l1=  
{ 'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8, 'J': 9, 'K': 10, 'L': 11}  
    s1=[]  
    s2=[]  
    s=[]  
    qing=[0]*12  
    zhong=[0]*12  
    for i in range(3):  
        shuru=input().split()  
        s1.append(shuru[0])  
        s2.append(shuru[1])  
        s.append(shuru[2])  
    for i in range(3):  
        if s[i]=='even':  
            for j in s1[i]:
```

```

        qing[l1[j]]=-114514
        zhong[l1[j]] = -114514
    for j in s2[i]:
        qing[l1[j]] = -114514
        zhong[l1[j]] = -114514
for i in range(3):
    if s[i]=='up':
        for j in s2[i]:
            qing[l1[j]]+=1
        for j in s1[i]:
            zhong[l1[j]]+=1
    if s[i]=='down':
        for j in s1[i]:
            qing[l1[j]]+=1
        for j in s2[i]:
            zhong[l1[j]]+=1
q=s.count('down')+s.count('up')
for i in range(12):
    if qing[i]==q:
        print(l[i], ' is the counterfeit coin and it is light.',sep='')
        break
    if zhong[i]==q:
        print(l[i], ' is the counterfeit coin and it is heavy.', sep='')
        break
n=int(input())
for i in range(n):
    main()

```

代码运行截图 (至少包含有"Accepted")

OpenJudge
题目ID, 标题, 描述
24n2400011257
信箱
账号

CS101 / 题库 (包括计概、数算题目)

[题目](#)
[排名](#)
[状态](#)
[提问](#)

#47792804提交状态

查看 提交 统计 提问

状态: **Accepted**

源代码

```

def main():
    l= {0:'A',1:'B',2:'C',3:'D',4:'E',5:'F',6:'G',7:'H',8:'I',9:'J',10:'K',11:'L'}
    l1= {'A':0,'B':1,'C':2,'D':3,'E':4,'F':5,'G':6,'H':7,'I':8,'J':9,'K':10,'L':11}
    s1=[]
    s2=[]
    s=[]
    qing=[0]*12
    zhong=[0]*12
    for i in range(3):
        shuru=input().split()
        s1.append(shuru[0])
        s2.append(shuru[1])
        s.append(shuru[2])
    for i in range(3):
        if s[i]=='even':

```

基本信息

#: 47792804

题目: 02692

提交人: 24n2400011257

内存: 3680kB

时间: 24ms

语言: Python3

提交时间: 2024-12-17 17:51:38

01088: 滑雪 (long long int time;)

dp, dfs similar, <http://cs101.openjudge.cn/practice/01088>

思路:

思路就是从地图每个格子开始遍历，然后dfs，用dp记录一下每个结点的最优解，便于在访问到的时候直接使用。

思路一开始就想到了，但是我在用代码实现的时候一直有问题，一直超时，最后问了ai才ac的。（问题出在我最开始dp的设定就不对，正确的dp只记录dfs后的最优解，我一直在更新dp表，难怪会超时，其实本质可能是我很久没做dfs了，使它成为了我最熟悉的陌生人了，过两天刷几道dfs吧，哎）

代码：

```
di=[(1,0),(0,1),(-1,0),(0,-1)]
from sys import setrecursionlimit
setrecursionlimit(10000)
m,n=map(int,input().split())
ma=[]
for i in range(m):
    ma.append(list(map(int,input().split())))
dp=[[0 for i in range(n)]for j in range(m)]
def dfs(x,y):
    if dp[x][y]:
        return dp[x][y]
    l=1
    for dx,dy in di:
        nx,ny=x+dx,y+dy
        if 0<=nx<m and 0<=ny<n and ma[x][y]<ma[nx][ny]:
            l1=dfs(nx,ny)+1
            l=max(l,l1)
    dp[x][y]=l
    return l
for i in range(m):
    for j in range(n):
        dfs(i,j)
ans=0
for i in dp:
    ans=max(ans,max(i))
print(ans)
```

代码运行截图 == (至少包含有"Accepted") ==

OpenJudge

题目ID, 标题, 描述

24n2400011257

信箱

账号

CS101 / 题库 (包括计概、数算题目)

题目

排名

状态

提问

#47798437提交状态

查看

提交

统计

提问

状态: Accepted

源代码

```
di=[(1,0),(0,1),(-1,0),(0,-1)]
from sys import setrecursionlimit
setrecursionlimit(10000)
m,n=map(int,input().split())
ma=[]
for i in range(m):
    ma.append(list(map(int,input().split())))
dp=[[0 for i in range(n)]for j in range(m)]
def dfs(x,y):
    if dp[x][y]:
        return dp[x][y]
    l=1
    for dx,dy in di:
        nx,ny=x+dx,y+dy
        if 0<=nx<m and 0<=ny<n and ma[x][y]<ma[nx][ny]:
            l=dfs(nx,ny)+1
```

基本信息

#: 47798437

题目: 01088

提交人: 24n2400011257

内存: 4468kB

时间: 40ms

语言: Python3

提交时间: 2024-12-17 21:30:18

25572: 螃蟹采蘑菇 (40min)

bfs, dfs, <http://cs101.openjudge.cn/practice/25572/>

思路:

我要疯了，这题debug了好久，一开始以为是算法错了，最后发现是输入读错了，我服了我就说为啥一直re啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊（还是leetcode友好）

就是两个格子的bfs，然后注意visited不要记录两个这样会出错，记录一个格子的状态就行了，应为一个格子定了，螃蟹的位置也就确定了

代码:

```
n=int(input())
from collections import deque
di=[(1,0),(0,1),(-1,0),(0,-1)]
ma=[]
sx1,sy1=-1,-1
for i in range(n):
    shuru=list(map(int,input().split()))
    if 9 in shuru:
        ex=i
        ey=shuru.index(9)
        shuru[ey]=0
    if 5 in shuru:
        for j in range(n):
            if shuru[j]==5:
                if sx1!=-1:
                    sx2 = i
                    sy2 = j
                    shuru[sy2]=0
                else:
                    sx1 = i
                    sy1=j
                    shuru[sy1]=0
        ma.append(shuru)
queue=deque()
v=[[1 for _ in range(n)]for _ in range(n)]
```

代码运行截图 (至少包含有"Accepted")

27373: 最大整数

代码：

```
m=int(input())
n=int(input())
num=list(map(str,input().split()))
for i in range(n):
    for j in range(n-1-i):
        if int(num[j+1]+num[j])<int(num[j]+num[j+1]):
            num[j],num[j+1]=num[j+1],num[j]
```

```

num1=[0]
num1.extend(num)
num=num1
dp=[['' for _ in range(1+m)]for _ in range(1+n)]
for i in range(1,1+n):
    if dp[i-1][len(num[i])]:
        if int(dp[i-1][len(num[i])])>int(num[i]):
            dp[i][len(num[i])]=dp[i-1][len(num[i])]
        else:
            dp[i][len(num[i])]=num[i]
    else:
        dp[i][len(num[i])]=num[i]
    for j in range(1,1+m):
        if dp[i-1][j]:
            if dp[i][j]:
                dp[i][j]=dp[i-1][j] if int(dp[i][j])<int(dp[i-1][j]) else dp[i-1][j]
            else:
                dp[i][j]=dp[i-1][j]
            if j+len(num[i])<=m:
                if dp[i][j+len(num[i])]:
                    dp[i][j+len(num[i])]=num[i]+dp[i-1][j] if int(num[i]+dp[i-1][j])>int(dp[i][j+len(num[i])]) else dp[i-1][j + len(num[i])]
                else:
                    dp[i][j + len(num[i])] = num[i]+dp[i - 1][j]
print(dp[-1][-1])

```

代码运行截图 (至少包含有"Accepted")

OpenJudge
题目ID, 标题, 描述
24n2400011257
信箱
账号

CS101 / 题库 (包括计概、数算题目)
题目
排名
状态
提问

#47764300提交状态
查看
提交
统计
提问

状态: Accepted

源代码

```

m=int(input())
n=int(input())
num=list(map(str,input().split()))
for i in range(n):
    for j in range(n-1-i):
        if int(num[j+1]+num[j])<int(num[j]+num[j+1]):
            num[j],num[j+1]=num[j+1],num[j]

num1=[0]
num1.extend(num)
num=num1
dp=[['' for _ in range(1+m)]for _ in range(1+n)]
for i in range(1,1+n):
    if dp[i-1][len(num[i])]:
        if int(dp[i-1][len(num[i])])>int(num[i]):
            dp[i][len(num[i])]=dp[i-1][len(num[i])]
        else:

```

基本信息
: 47764300
题目: 27373
提交人: 24n2400011257
内存: 31188kB
时间: 742ms
语言: Python3
提交时间: 2024-12-16 11:13:13

02811: 熄灯问题 (long long int time;)

brute force, <http://cs101.openjudge.cn/practice/02811>

思路:

其实slove函数是比较直观的（虽然我是在室友的提示下才知道的，但其实我感觉我玩一玩我发到群里的呢个链接也能想到这个局部解法）然后我一直卡在怎么解决最后一行剩的灯里，但其实根本不用解决。

（这就是我的思维还局限在数学上，而没有迁移到计算机上，因为此时已经可以暴力枚举了）

有个问题是slove函数根本没对第一行进行操作，那么我只需要枚举第一行的操作情况，然后再去用slove函数，直到slove函数解决了问题就输出它就行了。（比较破防的是我这个枚举不太会用递归去写，拿递归试了一会，红温了，就换成了ifelse暴力枚举）

代码：

```
from copy import deepcopy
m=[]
for _ in range(5):
    m.append(list(map(int,input().split())))
ma=deepcopy(m)
di=[(1,0),(0,1),(-1,0),(0,-1)]
ans=[[0]*6 for _ in range(5)]
def turn(i,j):
    global ma
    ma[i][j]=1-ma[i][j]
def change(i,j):
    global ans
    turn(i,j)
    ans[i][j]=1-ans[i][j]
    for dx,dy in di:
        nx,ny=i+dx,j+dy
        if 0<=nx<5 and 0<=ny<6:
            turn(nx,ny)
def solve():
    global ma,m,ans
    for i in range(1,5):
        for j in range(6):
            if ma[i-1][j]:
                change(i,j)
    if 1 in ma[-1]:
        ma=deepcopy(m)
        ans=[[0]*6 for _ in range(5)]
        return
    for i in ans:
        for j in i:
            print(j,end=' ')
        print()
    exit()
for i in range(64):
    if i%2==1:
        change(0,0)
    if (i%4)//2==1:
        change(0,1)
    if (i%8)//4==1:
        change(0,2)
    if (i%16)//8==1:
        change(0,3)
    if (i%32)//16==1:
        change(0,4)
    if (i%64)//32==1:
```

```
change(0,5)
solve()
```

代码运行截图 (至少包含有"Accepted")



OpenJudge 题目ID, 标题, 描述 24n2400011257 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#47807974提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
from copy import deepcopy
m=[]
for _ in range(5):
    m.append(list(map(int,input().split())))
ma=deepcopy(m)
di=[(1,0),(0,1),(-1,0),(0,-1)]
ans=[[0]*6 for _ in range(5)]
def turn(i,j):
    global ma
    ma[i][j]=1-ma[i][j]
def change(i,j):
    global ans
    turn(i,j)
    ans[i][j]=1-ans[i][j]
    for dx,dy in di:
        nx,ny=i+dx,j+dy
```

基本信息

#:	47807974
题目:	02811
提交人:	24n2400011257
内存:	3748kB
时间:	31ms
语言:	Python3
提交时间:	2024-12-18 10:54:28

08210: 河中跳房子

binary search, greedy, <http://cs101.openjudge.cn/practice/08210/>

思路:

先说一下我拿道题后的思路

一开始我还说这个题咋这么简单, 直接每次弹出最小的就行了, 结果。。。

上面思路错误的地方在于, 不一定是最小的, 比如间隔为4 3 的两个石头, 我弹出4后, 3变成7, 这样就缩短了两个最小路径, 如果我只弹出3, 后面还得再弹出一一次4, 就不是最优解了。

那么就知道了, 原来删除最小有两个方法 (直接去想很直观, 就是删除最小路径的两边的石头的任意一个), 那么我只需要把删除的最小路径, 加到它两边里面的更小的那一个路径。但是如果遇到4 3 4 3 这种, 在判断中间的3的时候, 应该往哪边加呢, 由于index永远记录的是最前面的那个3, 所以相等的情况下往前加就行了

直到我发现了 4 3 2 5 这组数据, 我就放弃了这个题是个greedy的想法, 然后搜题解, 找到了二分查找。。。 (二分查找本身不难理解, 就是不好往这个方向去想) 如果有greedy的题解一定要发一下, 我想学习学习

感觉二分查找真的是, 有点局限, 但对于这种题却又十分的好用, 积累一下。

代码:

```
l,n,m=map(int,input().split())
rocks=[0]
for i in range(n):
    rocks.append(int(input()))
rocks.append(1)
left=0
right=1
```


[illegible]

(感觉这一届同学更厉害了吧，我月考的水平只能在中等，但是做去年的题可以ac5，哎，优秀率退退

退!!!! 不要卡

[illegible][illegible][illegible][illegible][illegible]