

Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Compiled by 胡杨 元培学院

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

02692: 假币问题

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

思路：前一天晚上做了一个多小时搞不出来，今天受群里“枚举每枚硬币”和“正难则反”的启发

枚举每枚硬币，如果它在even组里出现，直接break。如果没有在even里出现过，如果它在轻的堆里，那么它一定不会是重的；如果在重的堆里，那么一定不会是轻的

但是有逻辑问题，如果硬币KJ在轻的堆里，然后又测试了一次K在轻的堆里，那K就是轻的，但是这个时候电脑无法判断是不是轻的，会输出两个。改不来了X(遂看题解

题解思路1：12枚硬币只有一枚假币，总共有24种情况：12枚硬币分别作为轻/重的假币，看是否符合题目给出的平衡条件。如果都符合，那么那一枚就是假币

这提醒我了cheating paper记得抄元组集合列表字典字符串的各种函数，实在是记不住啊

耗时：3天

代码：

```
n=int(input())
coins='ABCDEFGHIJKL'

def check(lines,coins):
    for line in lines:
        left,right,result=line
        left_weight=sum(weight[i] for i in left)
        right_weight=sum(weight[i] for i in right)

        if left_weight<right_weight and result!='down':
```

```

        return False
    if left_weight>right_weight and result!='up':
        return False
    if left_weight==right_weight and result!='even':
        return False
    return True

for _ in range(n):
    lines=[list(input().split()) for _ in range(3)]

    for i in coins:
        weight = {coin: 0 for coin in coins}
        found = False
        for w in [-1, 1]:
            weight[i] = w
            if check(lines,coins):
                found=True
                if w==-1:
                    print(f'{i} is the counterfeit coin and it is light.')
                elif w==1:
                    print(f'{i} is the counterfeit coin and it is heavy.')
                break

    if found:
        break

```

代码运行截图 (至少包含有"Accepted")

#47905198提交状态

状态: Accepted

源代码

```
n=int(input())
coins='ABCDEFGHIJKL'

def check(lines,coins):
    for line in lines:
        left,right,result=line
        left_weight=sum(weight[i] for i in left)
        right_weight=sum(weight[i] for i in right)

        if left_weight<right_weight and result!='down':
            return False
        if left_weight>right_weight and result!='up':
            return False
        if left_weight==right_weight and result!='even':
            return False
    return True

for _ in range(n):
    lines=[list(input().split()) for _ in range(3)]

    for i in coins:
        weight = {coin: 0 for coin in coins}
        found = False
        for w in [-1, 1]:
            weight[i] = w
```

01088: 滑雪

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

思路: 本来想要写一个dfs直接暴力遍历所有点的最大下滑距离取最大, 发现果然会超时, 除非使用缓存器

然后改为dp, 记录每一个点的最大下滑距离, 如果到了那个点直接调用该数值

耗时: 35min

记忆化搜索 (Memoization) 以避免对相同位置进行重复计算代码:

```
from functools import lru_cache

# 定义移动方向
dx = [0, 0, -1, 1]
dy = [1, -1, 0, 0]

@lru_cache(maxsize=None)
def dfs(x, y):
```

```

max_steps = 1 # 至少自己算一步
for i in range(4):
    nx, ny = x + dx[i], y + dy[i]
    if 0 <= nx < r and 0 <= ny < c and h[nx][ny] < h[x][y]:
        max_steps = max(max_steps, 1 + dfs(nx, ny))
return max_steps

# 读取输入
r, c = map(int, input().split())
h = []
for _ in range(r):
    h.append(tuple(map(int, input().split()))) # 使用tuple以便能被缓存

# 计算最大步数
max_step = 0
for i in range(r):
    for j in range(c):
        max_step = max(max_step, dfs(i, j))

print(max_step)

```

在这个版本中，我们做了以下改进：

1. **记忆化搜索**：使用 `@lru_cache` 装饰器来缓存已经计算过的结果，从而避免重复计算。
2. **输入处理**：将每一行的高度列表转换为元组（tuple），因为Python的 `lru_cache` 要求参数是可哈希的（hashable），而列表不是可哈希的，但元组是。

缓存器真好啊就是不知道oj有没有

dp手动缓存代码：

```

dx=[0,0,-1,1]
dy=[1,-1,0,0]

def dp1(x,y):
    if dp[x][y]!=-1:
        return dp[x][y]

    max_step=1
    for i in range(4):
        nx=x+dx[i]
        ny=y+dy[i]
        if 0<=nx<r and 0<=ny<c and h[nx][ny]<h[x][y]:
            max_step=max(max_step,dp1(nx,ny)+1)

    dp[x][y]=max_step
    return dp[x][y]

r,c=map(int,input().split())
h=[]
for _ in range(r):
    h.append(list(map(int,input().split())))
max_steps=1
dp=[[-1]*c for _ in range(r)]

```

```

for i in range(r):
    for j in range(c):
        max_steps=max(max_steps,dp1(i,j))

print(max_steps)

```

代码运行截图 (至少包含有"Accepted")

题目

排名

状态

提问

#47905783提交状态

状态: Accepted

源代码

```

dx=[0,0,-1,1]
dy=[1,-1,0,0]

def dp1(x,y):
    if dp[x][y]!=-1:
        return dp[x][y]

    max_step=1
    for i in range(4):
        nx=x+dx[i]
        ny=y+dy[i]
        if 0<=nx<r and 0<=ny<c and h[nx][ny]<h[x][y]:
            max_step=max(max_step,dp1(nx,ny)+1)

    dp[x][y]=max_step
    return dp[x][y]

r,c=map(int,input().split())
h=[]
for _ in range(r):
    h.append(list(map(int,input().split())))
max_steps=1
dp=[[-1]*c for _ in range(r)]

for i in range(r):

```

基本

提示

25572: 螃蟹采蘑菇

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

思路: bfs模板题, 占几个位置其实都是一样判断

耗时: 30min (码字太慢了啊啊啊啊, 要是可以有电子版cheating paper就好了(流口水))

代码:

```

from collections import deque
dx=[0,0,1,-1]
dy=[1,-1,0,0]

n=int(input())
maze=[]
for _ in range(n):
    line=list(map(int,input().split()))
    maze.append(line)

def bfs(a1,b1,a2,b2):
    q=deque()
    q.append(((a1,b1),(a2,b2)))
    inq=set()
    inq.add(((a1,b1),(a2,b2)))
    found=False
    while q:
        (x1,y1),(x2,y2)=q.popleft()
        if maze[x1][y1]==9 or maze[x2][y2]==9:
            found=True
            return found

        for i in range(4):
            nx1=x1+dx[i]
            nx2=x2+dx[i]
            ny1=y1+dy[i]
            ny2=y2+dy[i]
            if 0<=nx1<n and 0<=nx2<n and 0<=ny1<n and 0<=ny2<n and maze[nx1]
[ny1]!=1 and maze[nx2][ny2]!=1 and ((nx1,ny1),(nx2,ny2)) not in inq:
                q.append(((nx1,ny1),(nx2,ny2)))
                inq.add(((nx1,ny1),(nx2,ny2)))

    return found

start=[]
for i in range(n):
    for j in range(n):
        if maze[i][j]==5:
            start.append((i,j))
a1,b1,a2,b2=start[0][0],start[0][1],start[1][0],start[1][1]
result=bfs(a1,b1,a2,b2)
if result:
    print('yes')
else:
    print('no')

```

代码运行截图 (至少包含有"Accepted")

#47938025提交状态

状态: Accepted

源代码

```
from collections import deque
dx=[0,0,1,-1]
dy=[1,-1,0,0]

n=int(input())
maze=[]
for _ in range(n):
    line=list(map(int,input().split()))
    maze.append(line)

def bfs(a1,b1,a2,b2):
    q=deque()
    q.append(((a1,b1),(a2,b2)))
    inq=set()
    inq.add(((a1,b1),(a2,b2)))
    found=False
    while q:
        (x1,y1),(x2,y2)=q.popleft()
        if maze[x1][y1]==9 or maze[x2][y2]==9:
            found=True
            return found

        for i in range(4):
            nx1=x1+dx[i]
            nx2=x2+dx[i]
            ny1=y1+dy[i]
```

基本

提

提

27373: 最大整数

dp, <http://cs101.openjudge.cn/practice/27373/>

思路：首先需要对序列进行排序，使得每一个数添加在前面数的后面组成的数字是最大的，以便dp的时候能直接将正在考虑的数加在前面已经排好的数后面。dp[i]表示长度为i的最大整数，基本思路是对于每一个整数num，考虑选这个数int(str(dp[i-len(num)])+num)，或者不选这个数dp[i]的最大值

根据题解写的，并没有搞懂为什么反向更新可以避免重复

题目的难度和代码量看起来不一定成正比

耗时：1h

代码：

```
m=int(input())
n=int(input())
l=list(map(str,input().split()))

#将数字按照两个数连起来最大的方式排序
```

```

for i in range(n):
    for j in range(i+1,n):
        if int(l[i]+l[j])<int(l[j]+l[i]):
            l[i],l[j]=l[j],l[i]

dp=[0]*(m+1) #表示长度为i的最大整数
for num in l:
    for i in range(m,len(num)-1,-1): #反向更新避免重复(??)
        dp[i]=max(dp[i],int(str(dp[i-len(num)]+num)))

print(dp[m])

```

代码运行截图 (至少包含有"Accepted")



#47940554提交状态

状态: Accepted

基本信息

源代码

```

m=int(input())
n=int(input())
l=list(map(str,input().split()))

#将数字按照两个数连起来最大的方式排序
for i in range(n):
    for j in range(i+1,n):
        if int(l[i]+l[j])<int(l[j]+l[i]):
            l[i],l[j]=l[j],l[i]

dp=[0]*(m+1) #表示长度为i的最大整数
for num in l:
    for i in range(m,len(num)-1,-1): #反向更新避免重复(??)
        dp[i]=max(dp[i],int(str(dp[i-len(num)]+num)))

print(dp[m])

```

题目
提交
内容
时间
语言
提交时间



02811: 熄灯问题

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

思路:

代码:

代码运行截图 (至少包含有"Accepted")

08210: 河中跳房子

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

思路:

代码:

代码运行截图 (至少包含有"Accepted")

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。

脑壳剧痛中, 写完了四道遂先交一份

每次遇到难题就好崩溃, 永远都想不出来怎么做, 死磕两三天最后还是看答案, 然后也没时间做其他题了

cheat sheat 还没开始做, 我觉得现在更重要的应该是复习一下基础, 难题感觉属于一种碰运气成分。准备整理cheat sheat的时候也是复习的过程, 大概会包括列表集合等数据类型的基本用法, 我以前写过的一些不会的语法, dp dfs bfs的基本模板, 以及cheat一下同学的cheat sheat

期末求放过.....或许到考试的时候最重要的是我的头可以安好的不痛而清醒X(但基本都会痛X(