# Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 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

### 02692: 假币问题

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

思路：穷举，分别讨论每个币是轻的或者重的的情况，对三次称量进行判断，如果三次称量均与假设符合则假设正确，于是就找到了假币，需要注意的是有多组数据，所以找到假币break之前要先把dic[coin]重设为0，忘了这一步导致wa了半天。。。

代码：

```python

n=int(input())

dic={'A':0,'B':0,'C':0,'D':0,'E':0,'F':0,'G':0,'H':0,'I':0,'J':0,'K':0,'L':0}

coins=['A','B','C','D','E','F','G','H','I','J','K','L']

def judge(left,right,status):

a=b=0

for i in left:

a+=dic[i]

for i in right:

b+=dic[i]

if status=='even':

return a==b

if status=='down':

return a<b

if status=='up':

return a>b

for i in range(n):

left=[]

right=[]

status=[]

for j in range(3):

a,b,c=input().split()

left.append(a)

right.append(b)

status.append(c)

for coin in coins:

dic[coin]=1

if judge(left[0],right[0],status[0]) and judge(left[1],right[1],status[1]) and judge(left[2],right[2],status[2]):

print(f'{coin} is the counterfeit coin and it is heavy. ')

dic[coin]=0

break

dic[coin]=-1

if judge(left[0],right[0],status[0]) and judge(left[1],right[1],status[1]) and judge(left[2],right[2],status[2]):

print(f'{coin} is the counterfeit coin and it is light. ')

dic[coin]=0

break

dic[coin]=0

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

### 01088: 滑雪

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

思路：对于每一个点，如果四周的点更矮，那么这个点对应的长度就是四周点对应长度最大值+1，遍历四周的点即可

代码：

```python

R,C=map(int,input().split())

chemistry=[]

for i in range(R):

chemistry.append(list(map(int,input().split())))

jinhua=[[0]\*C for i in range(R)]

def dfs(x,y,chemistry,jinhua):

if jinhua[x][y]!=0:

return

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

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

for i in range(4):

nx,ny=x+dx[i],y+dy[i]

if 0<=nx<R and 0<=ny<C:

if chemistry[nx][ny]<chemistry[x][y]:

dfs(nx,ny,chemistry,jinhua)

jinhua[x][y]=max(jinhua[x][y],jinhua[nx][ny]+1)

if jinhua[x][y]==0:

jinhua[x][y]=1

for i in range(R):

for j in range(C):

dfs(i,j,chemistry,jinhua)

print(max(max(a) for a in jinhua))

```

代码运行截图 ==（至少包含有"Accepted"）==

### 25572: 螃蟹采蘑菇

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

思路：首先把起点终点找到，然后就跟传统dfs差不多，不过是两个点一起动，所以每个限制条件都得验两个点，注意到这两个点相对位置永远不变，所以visited只需要记录其中一个点到过的位置即可

代码：

```python

n=int(input())

maze=[]

for i in range(n):

maze.append(list(map(int,input().split())))

def mushroom(maze):

for i in range(n):

for j in range(n):

if maze[i][j]==9:

return (i,j)

def crab(maze):

for i in range(n):

for j in range(n):

if maze[i][j]==5:

for dir in [(0,1),(1,0)]:

if i+dir[0]<n and j+dir[1]<n and maze[i+dir[0]][j+dir[1]]==5:

return (i,j,i+dir[0],j+dir[1])

xe,ye=mushroom(maze)

x1,y1,x2,y2=crab(maze)

visited=[[False]\*n for i in range(n)]

can\_reach=False

def dfs(x1,y1,x2,y2,maze,visited):

global can\_reach

if (x1==xe and y1==ye) or (x2==xe and y2==ye):

can\_reach=True

return

if can\_reach:

return

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

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

for i in range(4):

nx1,ny1,nx2,ny2=x1+dx[i],y1+dy[i],x2+dx[i],y2+dy[i]

if 0<=nx1<n and 0<=ny1<n and 0<=nx2<n and 0<=ny2<n:

if not visited[nx1][ny1]:

if maze[nx1][ny1]!=1 and maze[nx2][ny2]!=1:

visited[x1][y1]=True

dfs(nx1,ny1,nx2,ny2,maze,visited)

dfs(x1,y1,x2,y2,maze,visited)

print('yes' if can\_reach else 'no')

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

### 27373: 最大整数

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

思路：没想出来，参考了题解

代码：

```python

m=int(input())

n=int(input())

num=list(input().split())

for i in range(n):

for j in range(n-1-i):

if num[j]+num[j+1]<num[j+1]+num[j]:

num[j],num[j+1]=num[j+1],num[j]

dp=[0]\*(m+1)

temp=[0]\*(m+1)

for i in num:

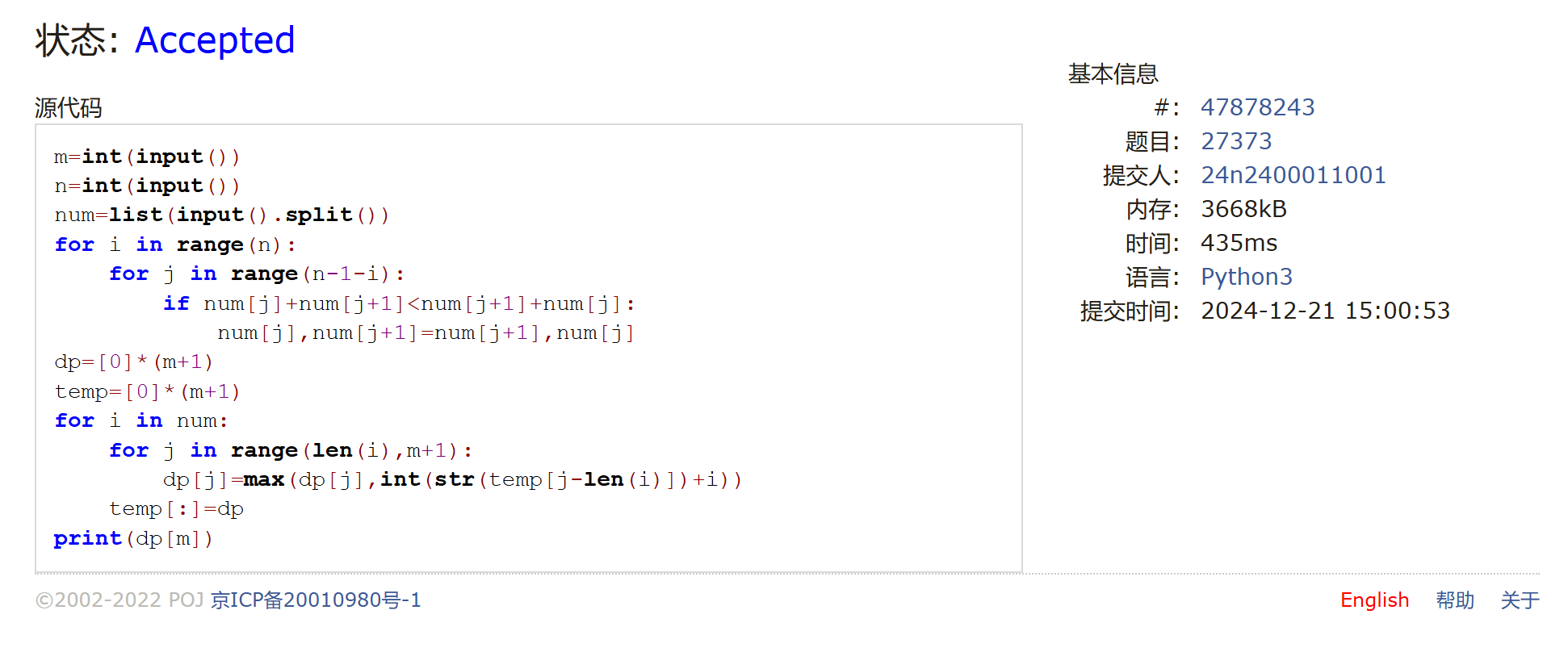
for j in range(len(i),m+1):

dp[j]=max(dp[j],int(str(temp[j-len(i)])+i))

temp[:]=dp

print(dp[m])

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

### 02811: 熄灯问题

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

思路：又是毫无头绪的一道题，这解法我想破头也想不出（哭

代码：

```python

import copy

light=[[0,0,0,0,0,0,0,0]]

for i in range(5):

light.append([0]+list(map(int,input().split()))+[0])

light.append([0,0,0,0,0,0,0,0])

button=[[0]\*8 for i in range(7)]

for a in range(2):

button[1][1]=a

for b in range(2):

button[1][2]=b

for c in range(2):

button[1][3]=c

for d in range(2):

button[1][4]=d

for e in range(2):

button[1][5]=e

for f in range(2):

button[1][6]=f

light1=copy.deepcopy(light)

button1=copy.deepcopy(button)

for i in range(1,7):

if button1[1][i]==1:

light1[1][i]=abs(light1[1][i]-1)

light1[1][i-1]=abs(light1[1][i-1]-1)

light1[1][i+1]=abs(light1[1][i+1]-1)

light1[2][i]=abs(light1[2][i]-1)

for i in range(2,6):

for j in range(1,7):

if light1[i-1][j]==1:

light1[i][j]=abs(light1[i][j]-1)

light1[i][j-1]=abs(light1[i][j-1]-1)

light1[i][j+1]=abs(light1[i][j+1]-1)

light1[i+1][j]=abs(light1[i+1][j]-1)

button1[i][j]=1

if light1[5][1]==light1[5][2]==light1[5][3]==light1[5][4]==light1[5][5]==light1[5][6]==0:

for i in range(1,6):

print(\*button1[i][1:7])

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

### 08210: 河中跳房子

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

思路：没想到tag里二分是要把跳跃距离二分（我想到的思路是每次操作把相邻的且和最小的两个距离合并，但是完全不知道怎么写成代码，还纳闷这和二分有啥关系……）

代码：

```python

L,N,M=map(int,input().split())

stone=[0]

for i in range(N):

stone.append(int(input()))

stone.append(L)

def check(stone,mid):

i=0

j=1

move=0

while j<=N+1:

if stone[j]-stone[i]<mid:

move+=1

j+=1

else:

i,j=j,j+1

return move<=M

l=0

r=L

ans=0

while l<r:

mid=(l+r)//2

if check(stone,mid):

l=mid+1

ans=mid

else:

r=mid

print(ans)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

## 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>

这次作业后三题太难了，都不会，没啥写下去的动力，基本就是看了看题解惊叹一下然后努力把答案默上去；练了练前两年的题，一个ac5一个ak（不过有的题做过），不知道今年难度和前两年比怎么样，期末目标保4争5，希望这次作业后三题那种难度的最多塞一道