

# Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Complied by 俞天麒 物理学院

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：

如果平衡，则把放上去的都标记为真。如果不平衡，那么就把没放上去的标记为真，高的一侧没有标记为真的标记一个可能为假，且轻，低的同理。直到最后没有标记为真的就是假币。

代码：

```
def trans(x):  
    return ord(x)-65  
n=int(input())  
for _ in range(n):  
    a=[-2]*12  
    for i in range(3):  
        x,y,z=input().split()  
        if z=="even":  
            for j in range(len(x)):  
                a[trans(x[j])],a[trans(y[j])]=0,0  
        else:  
            for j in range(12):  
                if chr(j+65) not in set(x+y):  
                    a[j]=0  
        if z=="up":  
            for j in range(len(x)):  
                if a[trans(x[j])]==2:  
                    a[trans(x[j])]=-1  
                elif a[trans(x[j])]==1:  
                    a[trans(x[j])]=0  
                if a[trans(y[j])]==2:  
                    a[trans(y[j])]=1
```

```

        elif a[trans(y[j])] == -1:
            a[trans(y[j])] = 0
    else:
        for j in range(len(x)):
            if a[trans(x[j])] == -2:
                a[trans(x[j])] = 1
            elif a[trans(x[j])] == -1:
                a[trans(x[j])] = 0
            if a[trans(y[j])] == -2:
                a[trans(y[j])] = -1
            elif a[trans(y[j])] == 1:
                a[trans(y[j])] = 0
    #print(a)
for i in range(12):
    if a[i] == 1:
        print(chr(i+65)+" is the counterfeit coin and it is light.")
        break
    elif a[i] == -1:
        print(chr(i+65)+" is the counterfeit coin and it is heavy.")
        break

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码	<pre> def trans(x):     return ord(x)-65 n=int(input()) for _ in range(n):     a=[-2]*12     for i in range(3):         x,y,z=input().split()         if z=="even":             for j in range(len(x)):                 a[trans(x[j])],a[trans(y[j])] = 0,0         else:             for j in range(12):                 if chr(j+65) not in set(x+y):                     a[j]=0             if z=="up":                 for j in range(len(x)):                     if a[trans(x[j])] == -2:                         a[trans(x[j])] = -1                     elif a[trans(x[j])] == 1:                         a[trans(x[j])] = 0                     if a[trans(y[j])] == -2:                         a[trans(y[j])] = 1                     elif a[trans(y[j])] == -1:                         a[trans(y[j])] = 0             else:                 for j in range(len(x)):                     if a[trans(x[j])] == -2:                         a[trans(x[j])] = 1                     elif a[trans(x[j])] == -1:                         a[trans(x[j])] = 0                     if a[trans(y[j])] == -2:                         a[trans(y[j])] = -1                     elif a[trans(y[j])] == 1:                         a[trans(y[j])] = 0     #print(a) for i in range(12):     if a[i]==1:         print(chr(i+65)+" is the counterfeit coin and it is light.")         break     elif a[i]==-1:         print(chr(i+65)+" is the counterfeit coin and it is heavy.")         break </pre>	<p>基本信息</p> <p>#: 46671920      题目: 02692      提交人: 24n2400011425      内存: 3712kB      时间: 21ms      语言: Python3      提交时间: 2024-10-23 09:15:35</p>
-----	--	---

# 01088: 滑雪

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

思路：

可以利用一个dp数组来储存已经访问过的值，标记为该点出发的最长下坡长度，简化计算。

代码：

```
r,c=map(int,input().split())
h=[list(map(int,input().split())) for i in range(r)]
length=[[float("-inf")]*c for i in range(r)]
dir=[[0,-1],[-1,0],[0,1],[1,0]]
def find(x,y):
    if length[x][y]!=float("-inf"):
        return length[x][y]
    ans=1
    for dx,dy in dir:
        nx,ny=x+dx,y+dy
        if 0<=nx<r and 0<=ny<c and h[nx][ny]<h[x][y]:
            ans=max(ans,find(nx,ny)+1)
    length[x][y]=ans
    return ans
a=1
for i in range(r):
    for j in range(c):
        a=max(a,find(i,j))
print(a)
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
r,c=map(int,input().split())
h=[list(map(int,input().split())) for i in range(r)]
length=[[float("-inf")]*c for i in range(r)]
dir=[[0,-1],[-1,0],[0,1],[1,0]]
def find(x,y):
    if length[x][y]!=float("-inf"):
        return length[x][y]
    ans=1
    for dx,dy in dir:
        nx,ny=x+dx,y+dy
        if 0<=nx<r and 0<=ny<c and h[nx][ny]<h[x][y]:
            ans=max(ans,find(nx,ny)+1)
    length[x][y]=ans
    return ans
a=1
for i in range(r):
    for j in range(c):
        a=max(a,find(i,j))
print(a)
```

基本信息

#: 47785912  
题目: 01088  
提交人: 24n2400011425  
内存: 4376kB  
时间: 46ms  
语言: Python3  
提交时间: 2024-12-17 13:46:43

## 25572: 螃蟹采蘑菇

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

思路：

相当于在常规的bfs上附加一个相邻格点的判断

代码：

```
from collections import deque
n=int(input())
a=[list(map(int,input().split())) for i in range(n)]
dir=[[0,1],[0,-1],[1,0],[-1,0]]
t=[[0,1],[1,0]]
visited=[[True]*n for i in range(n)]
sx,sy=-1,-1
ty=0
for i in range(n):
    for j in range(n):
        if a[i][j]==5:
            sx,sy=i,j
            if 0<=sx<=n-1 and a[sx+1][sy]==5:
                ty=1
                break
        if sx!=-1:
            break
def find(x,y):
    global ty
    visit=deque([(x,y)])
    while visit:
        sx,sy=visit.popleft()
        #print(sx,sy,end="!")
        visited[sx][sy]=False
        x1,y1=sx+t[ty][0],sy+t[ty][1]
        if a[sx][sy]==9 or a[x1][y1]==9:
            return "yes"
        else:
            for dx,dy in dir:
                nx,ny=sx+dx,sy+dy
                #print(nx,ny)
                x2,y2=nx+t[ty][0],ny+t[ty][1]
                if 0<=nx<=n and 0<=ny<=n and a[nx][ny]!=1 and 0<=x2<=n and 0<=y2<=n and a[x2][y2]!=1 and visited[nx][ny]:
                    visit.append((nx,ny))
    return "no"
print(find(sx,sy))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import deque
n=int(input())
a=[list(map(int,input().split())) for i in range(n)]
dir=[[0,1],[0,-1],[1,0],[-1,0]]
t=[[0,1],[1,0]]
visited=[[True]*n for i in range(n)]
sx,sy=-1,-1
ty=0
for i in range(n):
    for j in range(n):
        if a[i][j]==5:
            sx,sy=i,j
            if 0<=sx<=n-1 and a[sx+1][sy]==5:
                ty=1
                break
            break
        if sx!=-1:
            break
def find(x,y):
    global ty
    visit=deque([(x,y)])
    while visit:
        sx,sy=visit.popleft()
        #print(sx,sy,end="")
        visited[sx][sy]=False
        x1,y1=sx+t[ty][0],sy+t[ty][1]
        if a[sx][sy]==9 or a[x1][y1]==9:
            return "yes"
        else:
            for dx,dy in dir:
                nx,ny=sx+dx,sy+dy
                #print(nx,ny)
                x2,y2=nx+t[ty][0],ny+t[ty][1]
                if 0<=nx<=n and 0<=ny<=n and a[nx][ny]!=1 and 0<=x2<=n and
                    visit.append((nx,ny))
    return "no"
print(find(sx,sy))
```

基本信息

#: 47786215  
题目: 25572  
提交人: 24n2400011425  
内存: 3712kB  
时间: 22ms  
语言: Python3  
提交时间: 2024-12-17 14:04:14

## 27373: 最大整数

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

思路:

先用greedy确定了在同样位数的情况下什么数放前面更大，之后利用小偷背包解决只能放m位的问题。

代码:

```
m=int(input())
n=int(input())
a=list(map(int,input().split()))
def sort_key(x):
    return x/(10***(len(str(x))-1))
a.sort(key=sort_key,reverse=True)
a=list(map(str,a))
#print(a)
dp=[0]*(m+1)
for i in range(n):
    for j in range(m,len(a[i])-1,-1):
        dp[j]=max(dp[j],int(str(dp[j-len(a[i])])+a[i]))
#print(dp)
print(max(dp))
```

## 代码运行截图 (至少包含有"Accepted")

#47787117提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
m=int(input())
n=int(input())
a=list(map(int,input().split()))
def sort_key(x):
    return x/(10***(len(str(x))-1))
a.sort(key=sort_key,reverse=True)
a=list(map(str,a))
#print(a)
dp=[0]*(m+1)
for i in range(n):
    for j in range(m,len(a[i])-1,-1):
        dp[j]=max(dp[j],int(str(dp[j-len(a[i])])+a[i]))
#print(dp)
print(max(dp))
```

©2002-2022 POJ 京ICP备20010980号-1

基本信息

#: 47787117  
题目: 27373  
提交人: 24n2400011425  
内存: 3732kB  
时间: 271ms  
语言: Python3  
提交时间: 2024-12-17 14:52:43

English 帮助 关于

## 02811: 熄灯问题

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

思路:

没想到真的是要去暴力枚举第一行，还以为有什么一次性的方法

代码:

```
a=[list(map(int,input().split())) for i in range(5)]
record=[]
for i in range(5):
    temp=[]
    for j in range(6):
        temp.append(a[i][j])
    record.append(temp)
man=[[0]*6 for i in range(5)]
dir=[[0,0],[0,1],[1,0],[0,-1],[-1,0]]
def do(x,y):
    man[x][y]=1
    for dx,dy in dir:
        nx,ny=x+dx,y+dy
        if 0<=nx<5 and 0<=ny<6:
            if a[nx][ny]==0:
                a[nx][ny]=1
            else:
                a[nx][ny]=0
def make(n):
    ans=[]
    temp=[]
    def inmake(i):
        #print(i,temp)
        if i==n:
            temp1=[temp[i] for i in range(len(temp))]
            ans.append(temp1)
            return
        for j in range(6):
            if a[i][j]==0:
                temp.append(j)
                inmake(i+1)
                temp.pop()
    inmake(0)
    return ans
```

```

    else:
        for j in range(2):
            temp.append(j)
            inmake(i+1)
            temp.pop()

    inmake(0)
    return ans

perm=make(6)
for element in perm:
    for i in range(6):
        if element[i]==1:
            do(0,i)
    for i in range(4):
        for j in range(6):
            if a[i][j]==1:
                do(i+1,j)
    if a[4].count(1)==0:
        for i in range(5):
            print(" ".join(map(str,man[i])))
        break
    else:
        man=[[0]*6 for i in range(5)]
        for i in range(5):
            for j in range(6):
                a[i][j]=record[i][j]

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

a=[list(map(int,input().split())) for i in range(5)]
record=[]
for i in range(5):
    temp=[]
    for j in range(6):
        temp.append(a[i][j])
    record.append(temp)

man=[[0]*6 for i in range(5)]
dir=[[0,0],[0,1],[1,0],[0,-1],[-1,0]]
def do(x,y):
    man[x][y]=1
    for dx,dy in dir:
        nx,ny=x+dx,y+dy
        if 0<=nx<5 and 0<=ny<6:
            if a[nx][ny]==0:
                a[nx][ny]=1
            else:
                a[nx][ny]=0

def make(n):
    ans=[]
    temp=[]
    def inmake(i):
        #print(i,temp)
        if i==n:
            templ=[temp[i] for i in range(len(temp))]
            ans.append(templ)
            return
        else:
            for j in range(2):
                temp.append(j)
                inmake(i+1)
                temp.pop()

    inmake(0)
    return ans

perm=make(6)

```

基本信息

#: 47788150  
 题目: 02811  
 提交人: 24n2400011425  
 内存: 3752kB  
 时间: 23ms  
 语言: Python3  
 提交时间: 2024-12-17 15:47:11

## 08210: 河中跳房子

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

思路：

答案的思路很妙，原本我以为二分是要用二分的方法去寻找现在石头堆里最短的距离，然后去除相应的石头，但是这样答案不对，不知道为什么。看了答案才知道原来可以用二分来寻找距离。

代码：

```
l,n,m=map(int,input().split())
a=[0]
for i in range(n):
    a.append(int(input()))
a.append(l)
def rock(x):
    global m
    cur,num=0,0
    for i in range(1,n+2):
        if a[i]-cur<x:
            num+=1
        else:
            cur=a[i]
    if num>m:
        return True
    else:
        return False
left=0
right=l
ans=0
while left<right:
    mid=(left+right)//2
    if rock(mid):
        right=mid
    else:
        ans=mid
        left=mid+1
print(ans)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
l,n,m=map(int,input().split())
a=[0]
for i in range(n):
    a.append(int(input()))
a.append(1)
def rock(x):
    global m
    cur,num=0,0
    for i in range(l,n+2):
        if a[i]-cur<=x:
            num+=1
        else:
            cur=a[i]
    if num>m:
        return True
    else:
        return False
left=0
right=l
ans=0
while left<right:
    mid=(left+right)//2
    if rock(mid):
        right=mid
    else:
        ans=mid
        left=mid+1
print(ans)
```

©2007-2022 POJ 吉TCPOJ20010980号-1

基本信息

#: 47789183  
题目: 08210  
提交人: 24n2400011425  
内存: 5580kB  
时间: 253ms  
语言: Python3  
提交时间: 2024-12-17 16:13:53

English 帮助 羊干

## 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。

好多题目要看了答案才会做，但是这样就不知道自己水平有没有长进了。不知道这样和一个无情的背答案机器有什么区别。这样子的学习状态正常吗？理想中学习的难道不应该是自己想出来，现在只能一味的依赖答案，自己要不是想不出来，就是做法太复杂。不知道这样子算不算畏难情绪，会不会在考试中吃亏。