

Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Compiled by <mark>祁黄奕 城市与环境学院</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：创建存储状态信息的 dic 字典和包含所有硬币编号的 total 集合。
locate 函数通过 split 来判断硬币位置。遍历称量结果，将平衡时的硬币放入 normal 集。计算出可能为假币的 uncertain 集。对于 uncertain 中的硬币，遍历称量结果，若非平衡，通过 locate 找位置并计算状态，根据状态首次赋值或判断一致性，不一致则标记，未标记的为假币。

代码：

```
```python
```

```
def weigh(s1, s2, s3):
 dic = {'left': -1, 'right': 1, 'up': 1, 'down': -1, 1:
'light', -1: 'heavy'}
```

```

total = set('ABCDEFGHijkl')
normal = set()
def locate(x, m):
 left, right, _ = m.split()
 if x in left:
 return 'left'
 elif x in right:
 return 'right'
 else:
 return 0
for i in (s1, s2, s3):
 if i.split()[-1] == 'even':
 for j in i:
 if 'A' <= j <= 'L':
 normal.add(j)
uncertain = total - normal
for i in uncertain:
 heavy_light = None
 break_flag = False
 for j in (s1, s2, s3):
 left, right, result = j.split()
 if result == 'even':
 continue
 location = locate(i, j)
 if location == 0:
 break_flag = True
 break
 res = dic[location] * dic[result]
 if heavy_light is None:
 heavy_light = dic[res]
 elif heavy_light != dic[res]:
 break_flag = True
 break
 if not break_flag:
 return f'{i} is the counterfeit coin and it is
{heavy_light}.'

n = int(input())
ans = []
for _ in range(n):
 a = str(input()).rstrip()
 b = str(input()).rstrip()
 c = str(input()).rstrip()

```

```

 ans.append(weigh(a, b, c))
print(*ans, sep='\n')
'''

```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

#47925464提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

def weigh(s1, s2, s3):
 dic = {'left': -1, 'right': 1, 'up': 1, 'down': -1, 1: 'light', -1: 'heavy'}
 total = set('ABCDEFGHJKL')
 normal = set()
 def locate(x, m):
 left, right, _ = m.split()
 if x in left:
 return 'left'
 elif x in right:
 return 'right'

```

基本信息

#: 47925464  
 题目: 02692  
 提交人: 2400013403  
 内存: 3660kB  
 时间: 30ms  
 语言: Python3  
 提交时间: 2024-12-23 23:01:54

### 01088: 滑雪

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

思路: 用 dfs 搜索找矩阵中最长递增路径, 采用记忆化搜索, 在 dfs 里先判断是否已算过。

代码:

```python

```

r, c = map(int, input().split())
matrix = [list(map(int, input().split())) for _ in range(r)]
dp = [[0 for _ in range(c)] for _ in range(r)]

def dfs(x, y):
    if dp[x][y] != 0:
        return dp[x][y]
    directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]
    max_path = 1
    for dx, dy in directions:
        new_x, new_y = x + dx, y + dy
        if 0 <= new_x < r and 0 <= new_y < c and matrix[x][y] >
matrix[new_x][new_y]:
            path_length = dfs(new_x, new_y) + 1
            max_path = max(max_path, path_length)
    dp[x][y] = max_path
    return max_path

ans = 0
for row in range(r):
    for col in range(c):

```

```

        ans = max(ans, dfs(row, col))
    print(ans)
'''

```

代码运行截图 ==（至少包含有“Accepted”）==

#47925821提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

r, c = map(int, input().split())
matrix = [list(map(int, input().split())) for _ in range(r)]
dp = [[0 for _ in range(c)] for _ in range(r)]

def dfs(x, y):
    if dp[x][y] != 0:
        return dp[x][y]

```

基本信息

#: 47925821
 题目: 01088
 提交人: 2400013403
 内存: 4316kB
 时间: 43ms
 语言: Python3
 提交时间: 2024-12-23 23:26:35

25572: 螃蟹采蘑菇

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

思路：先读取迷宫信息及小呆起始坐标，计算其两部分坐标差值。通过定义合法位置判断函数，利用栈模拟小呆移动探索，按四个方向尝试。

代码：

```python

```

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

for i in range(n):
 for j in range(n):
 if maze[i][j] == 5:
 start.append((i, j))

delta_x = start[1][0] - start[0][0]
delta_y = start[1][1] - start[0][1]
visited = set()
visited.add((start[0][0], start[0][1]))

def is_valid(r, c):
 return (0 <= r < n and 0 <= c < n and (r, c) not in visited
 and maze[r][c] != 1 and 0 <= r + delta_x < n and 0 <= c
 + delta_y < n
 and maze[r + delta_x][c + delta_y] != 1)

dx = [0, 0, 1, -1]
dy = [1, -1, 0, 0]
stack = [(start[0][0], start[0][1])]

```

```

while stack:
 front_x, front_y = stack.pop()

 if maze[front_x][front_y] == 9 or maze[front_x +
delta_x][front_y + delta_y] == 9:
 print('yes')
 break
 for i in range(4):
 new_x, new_y = front_x + dx[i], front_y + dy[i]
 if is_valid(new_x, new_y):
 visited.add((new_x, new_y))
 stack.append((new_x, new_y))
else:
 print('no')
'''

```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

#47925964提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

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

for i in range(n):
 for j in range(n):
 if maze[i][j] == 5:
 start.append((i, j))

```

基本信息

#: 47925964  
 题目: 25572  
 提交人: 2400013403  
 内存: 3712kB  
 时间: 23ms  
 语言: Python3  
 提交时间: 2024-12-23 23:36:34

### 27373: 最大整数

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

思路: 先对数据进行预处理, 再进行 dp, 由于一个数只能用一次, 所以添加了一个 temp 来缓存。

代码:

```python

```

limit=int(input())
n=int(input())
lst=sorted(list(map(str,input().split())) ,reverse=True)

def sorting(a,b):
    if int(a+b)>int(b+a):
        return True
    else:
        return False
for i in range(n):
    for j in range(i,n):

```

```

        if not sorting(lst[i],lst[j]):
            lst[i],lst[j]=lst[j],lst[i]

dp=[0]*(limit+1)
temp=[0]*(limit+1)
for j in lst:
    for i in range(len(j),limit+1):
        dp[i]=max(dp[i],int(str(temp[i-len(j)]))+j)
    temp[:]=dp
print(dp[limit])
```

```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

#47926270提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

limit=int(input())
n=int(input())
lst=sorted(list(map(str,input().split())),reverse=True)

def sorting(a,b):
 if int(a+b)>int(b+a):
 return True

```

基本信息

#: 47926270

题目: 27373

提交人: 2400013403

内存: 3976kB

时间: 549ms

语言: Python3

提交时间: 2024-12-24 00:05:35

### 02811: 熄灯问题

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

思路：遍历第一行所有可能，再根据提示解决二到五行。

代码：

```

```python
from copy import deepcopy

matrix=[list(map(int,input().split())) for _ in range(5)]
target=[[0 for _ in range(6)] for _ in range(5)]

def change(x,y):
    if cnt[x][y]==1:
        cnt[x][y]=0
    else:
        cnt[x][y]=1

def turn(x,y):
    dx=[-1,1,0,0,0]
    dy=[0,0,-1,1,0]
    for o in range(5):

```


代码:

```
```python
end,n,limit=map(int,input().split())
pre=0
delta=[]
for _ in range(n):
 x=int(input())
 delta.append(x-pre)
 pre=x
delta.append(end-pre)

def check(distance):
 num=0
 cumulative=0
 for i in range(n+1):
 if delta[i]+cumulative>=distance:
 cumulative=0
 continue
 cumulative+=delta[i]
 num+=1
 if num<=limit:
 return True
 else:
 return False

low=0
high=end
mid=0
while low<high-1:
 mid=(low+high)//2
 if check(mid):
 low=mid
 else:
 high=mid
if check(mid):
 print(mid)
else:
 print(mid-1)
```
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: **Accepted**

源代码

```
end,n,limit=map(int,input().split())
pre=0
delta=[]
for _ in range(n):
    x=int(input())
    delta.append(x-pre)
    pre=x
```

基本信息

#: 47926385
题目: 08210
提交人: 2400013403
内存: 5616kB
时间: 300ms
语言: Python3
提交时间: 2024-12-24 00:20:22

2. 学习总结和收获

这次作业每一题都难啊难难难！马上就上机考试了真的求善良求放过呜呜……且
笔试要记的内容也好多呀……依然在做最后的努力（垂死挣扎），疑似失去所有
力气和手段。