

# Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Compiled by 洪千濠 工学院

## 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：耗时20分钟

代码：

```
s="ABCDEFGHIJKL"
for _ in range(int(input())):
    p=[list(input().split()) for _ in range(3)]
    for i in s:
        if all((i in j[0] and j[2]=="up") or (i in j[1] and j[2]=="down") or (i
not in j[1] and i not in j[0] and j[2]=="even") for j in p):
            print(f"{i} is the counterfeit coin and it is heavy.")
            break
        elif all((i in j[0] and j[2]=="down") or (i in j[1] and j[2]=="up") or (i
not in j[1] and i not in j[0] and j[2]=="even") for j in p):
            print(f"{i} is the counterfeit coin and it is light.")
            break
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
s="ABCDEFGHIJKL"
for _ in range(int(input())):
    p=list(input().split())
    for i in range(3):
        if all((i in j[0] and j[2]=="up") or (i in j[1] and j[2]=="down")):
            print(f"{i} is the counterfeit coin and it is heavy.")
            break
        elif all((i in j[0] and j[2]=="down") or (i in j[1] and j[2]=="up")):
            print(f"{i} is the counterfeit coin and it is light.")
            break
```

基本信息

#: 47893027  
题目: 02692  
提交人: 24n2400011114  
内存: 3948kB  
时间: 24ms  
语言: Python3  
提交时间: 2024-12-22 13:20:50

## 01088: 滑雪

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

思路: 耗时30分钟

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):
    dx = [0, 0, 1, -1]
    dy = [1, -1, 0, 0]
    for i in range(4):
        nx, ny = x + dx[i], y + dy[i]
        if 0 <= nx < r and 0 <= ny < c and matrix[x][y] > matrix[nx][ny]:
            if dp[nx][ny] == 0:
                dfs(nx, ny)
            dp[x][y] = max(dp[x][y], dp[nx][ny] + 1)
    if dp[x][y] == 0:
        dp[x][y] = 1

ans = 0
for o in range(r):
    for j in range(c):
        if not dp[o][j]:
            dfs(o, j)
    ans = max(ans, dp[o][j])
print(ans)
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

基

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):
    dx = [0, 0, 1, -1]
    dy = [1, -1, 0, 0]
    for i in range(4):
        nx, ny = x + dx[i], y + dy[i]
        if 0 <= nx < r and 0 <= ny < c and matrix[x][y] > matrix[nx][ny]:
            if dp[nx][ny] == 0:
                dfs(nx, ny)
```

## 25572: 螃蟹采蘑菇

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

思路：直接学习了题解，原来的代码太混乱。

代码：

```
from collections import deque

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):
    if 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:
        return 1
    else:
        return 0

dx=[0,0,1,-1]
dy=[1,-1,0,0]
stack=deque()
stack.append((start[0][0],start[0][1]))
while stack:
    front_x,front_y=stack.popleft()
    if maze[front_x][front_y]==9 or maze[front_x+delta_x][front_y+delta_y]==9:
        print('yes')
```

```

        exit()
    for i in range(4):
        nx,ny=front_x+dx[i],front_y+dy[i]
        if is_valid(nx,ny):
            visited.add((nx,ny))
            stack.append((nx,ny))
    print('no')

```

代码运行截图 (至少包含有"Accepted")

## #47893111提交状态

状态: Accepted

源代码

```

from collections import deque

```

## 27373: 最大整数

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

思路: 耗时30分钟

代码:

```

import math
limit = int(input())
n = int(input())
lst = list(map(str, input().split()))
max_len = len(max(lst, key = lambda x:len(x)))
lst.sort(key = lambda x: x * math.ceil(2*max_len/len(x)), reverse = True)
dp = [0] * (limit + 1)
for num in lst:
    for i in range(limit, len(num) - 1, -1):
        dp[i] = max(dp[i], dp[i - len(num)] * (10 ** len(num)) + int(num))
print(dp[limit])

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

主

源代码

```
import math
limit = int(input())
n = int(input())
lst = list(map(str, input().split()))
max_len = len(max(lst, key = lambda x:len(x)))
lst.sort(key = lambda x: x * math.ceil(2*max_len/len(x)), reverse = True)
dp = [0] * (limit + 1)
```

## 02811: 熄灯问题

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

思路: 自己看题解之后复写了一遍, 还是不太懂, 再看看www

代码:

```
from copy import deepcopy
from itertools import product
rmap = {0:1, 1:0}
matrix_backup = [[0] * 8] + [[0, *map(int, input().split()), 0] for i in
range(5)] \
    + [[0] * 8]

for test in product(range(2), repeat=6):
    matrix = deepcopy(matrix_backup)
    triggers = [list(test)]
    for i in range(1, 6):
        for j in range(1, 7):
            if triggers[i - 1][j - 1]:
                matrix[i][j] = rmap[matrix[i][j]]
                matrix[i - 1][j] = rmap[matrix[i - 1][j]]
                matrix[i + 1][j] = rmap[matrix[i + 1][j]]
                matrix[i][j - 1] = rmap[matrix[i][j - 1]]
                matrix[i][j + 1] = rmap[matrix[i][j + 1]]
            triggers.append(matrix[i][1:7])
    if matrix[5][1:7] == [0, 0, 0, 0, 0, 0]:
        for trigger in triggers[:-1]:
            print(' '.join(map(str, trigger)))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from copy import deepcopy
from itertools import product
rmap = {0:1, 1:0}
matrix_backup = [[0] * 8] + [[0, *map(int, input().split()), 0] for i in range(6)] + [[0] * 8]

for test in product(range(2), repeat=6):
    matrix = deepcopy(matrix_backup)
    triggers = [list(test)]
    for i in range(1, 6):
        for j in range(1, 7):
            if triggers[i - 1][j - 1]:
```

## 08210: 河中跳房子

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

思路: 耗时40分钟

代码:

```
L,n,m = map(int,input().split())
rock = [0]
for i in range(n):
    rock.append(int(input()))
rock.append(L)

def check(x):
    num = 0
    now = 0
    for i in range(1, n+2):
        if rock[i] - now < x:
            num += 1
        else:
            now = rock[i]

    if num > m:
        return True
    else:
        return False

lo, hi = 0, L
ans = -1
while lo < hi:
    mid = (lo + hi) // 2

    if check(mid):
        hi = mid
    else:
        ans = mid
        lo = mid + 1
print(ans)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

1

源代码

```
L,n,m = map(int,input().split())
rock = [0]
for i in range(n):
    rock.append(int(input()))
rock.append(L)

def check(x):
    num = 0
    now = 0
    for i in range(1, n+2):
```

## 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。

积极备战，希望机考温柔以待。