# Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Complied by <mark>付耀贤 信息管理系</mark>

## 1. 题目

### 02692: 假币问题

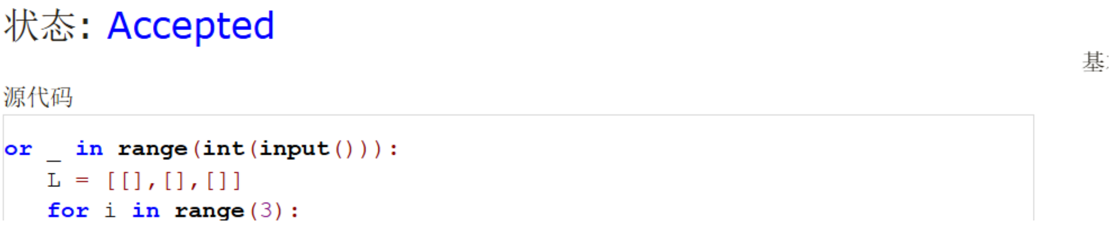
brute force, http://cs101.openjudge.cn/practice/02692

思路：不好做，一道在处理和思路上都谈不上很容易的题目。有点思路，但做着做着就迷糊了，自己被绕进去了：写了1000+B，还是WA。题解还是一如既往地牛。只能学习题解，希望提高点思维能力，，

代码：

```python
for _ in range(int(input())):
    L = [[],[],[]]
    for i in range(3):
        L[i] = input().split()
    for f in 'ABCDEFGHIJKL':
        if all((f in i[0] and i[2]=='up') or (f in i[1] and i[2]=='down')
                  or ( f not in i[0] + i[1] and i[2]=='even') for i in L):
            print("{} is the counterfeit coin and it is {}.".format(f,'heavy'))
            break
        if all((f in i[0] and i[2]=='down') or (f in i[1] and i[2]=='up')
                  or (f not in i[0]+i[1] and i[2]=='even') for i in L):
            print("{} is the counterfeit coin and it is {}.".format(f,'light'))
            break
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



状态: Accepted

基:

源代码

```
or _ in range(int(input())):
   L = [[],[],[]]
   for i in range(3):
```

### 01088: 滑雪

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

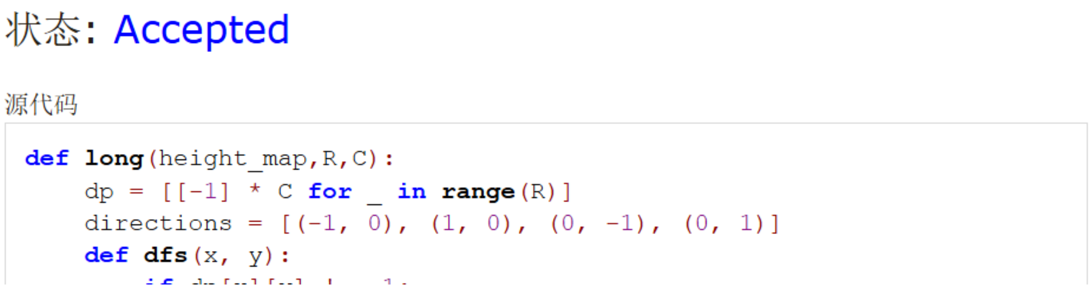
思路：dfs+dp。虽然 AI 辅助，但也是基本厘清思路，做出来了，有点激动。

代码：

```
def long(height_map,R,C):
    dp = [[-1] * C for _ in range(R)]
    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]
    def dfs(x, y):
        if dp[x][y] != -1:
            return dp[x][y]
        m = 1
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < R and 0 <= ny < C and height_map[nx][ny]
< height_map[x][y]:
                m = max(m, 1 + dfs(nx, ny))
        dp[x][y] = m
        return m
    max_len = 0
    for i in range(R):
        for j in range(C):
            max_len = max(max_len, dfs(i, j))
    return max_len

R, C = map(int, input().split())
height_map = [list(map(int, input().split())) for _ in range(R)]
result = long(height_map,R,C)
print(result)
```

代码运行截图 == （至少包含有"Accepted"） ==



状态: Accepted

源代码

```
def long(height_map,R,C):
    dp = [[-1] * C for _ in range(R)]
    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]
    def dfs(x, y):
        if dp[x][y] != -1:
```

### 25572: 螃蟹采蘑菇

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

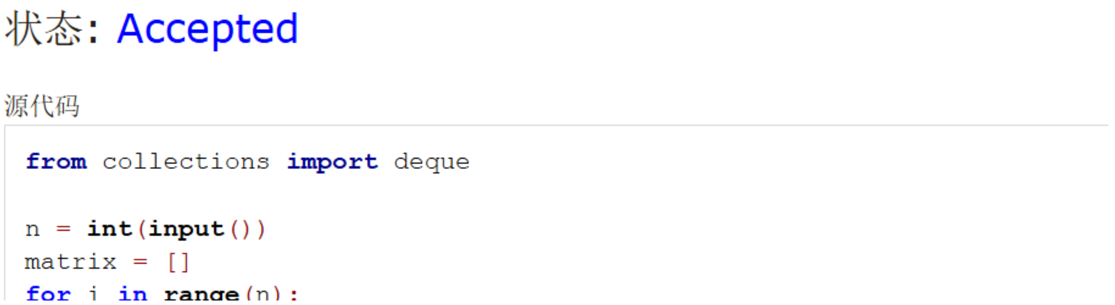
思路：这是某年的期末题，当时做的时候直接跳过了，后来再看还是不会。小螃蟹再瘦点，只占用 1 个格子多好，这样就是模板题，我就会做了（笑。题解的方法能看懂，但写不出来

代码：

```
from collections import deque

n = int(input())
matrix = []
for i in range(n):
    matrix.append(list(map(int, input().split())))
a = []
for i in range(n):
    for j in range(n):
        if matrix[i][j] == 5:
            a.append([i, j])
lx = a[1][0] - a[0][0]
ly = a[1][1] - a[0][1]
dire = [[-1, 0], [0, 1], [1, 0], [0, -1]]
v = [[0] * n for i in range(n)]
def bfs(x, y):
    v[x][y] = 1
    que = deque([(x, y)])
    while que:
        x, y = que.popleft()
        if (matrix[x][y] == 9 and matrix[x + lx][y + ly] != 1) or 
\
                (matrix[x][y] != 1 and matrix[x + lx][y + ly] ==
9):
            return 'yes'
        for i in range(4):
            dx = x + dire[i][0]
            dy = y + dire[i][1]
            if 0 <= dx < n and 0 <= dy < n and 0 <= dx + lx < n \
                    and 0 <= dy + ly < n and v[dx][dy] == 0 \
                    and matrix[dx][dy] != 1 and matrix[dx +
lx][dy + ly] != 1:
                que.append([dx, dy])
                v[dx][dy] = 1
    return 'no'
print(bfs(a[0][0], a[0][1]))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



状态: Accepted

源代码

```
from collections import deque

n = int(input())
matrix = []
for i in range(n):
```

### 27373: 最大整数

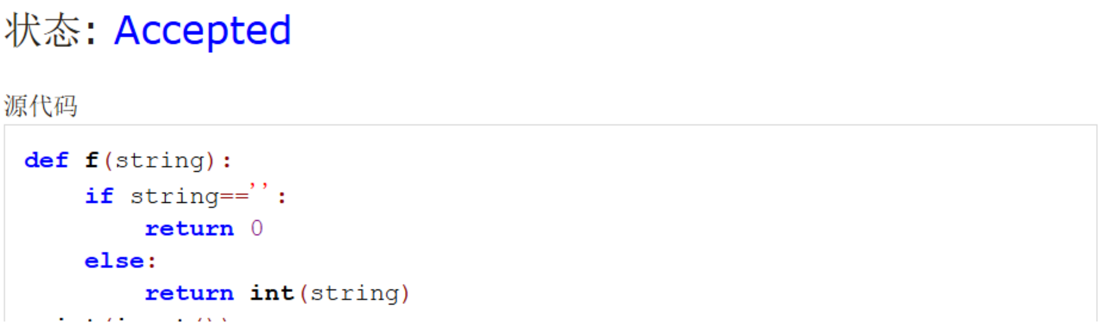
dp, http://cs101.openjudge.cn/practice/27373/

思路：好题，难题。这道题我的朴素直觉是在 DP 中使用两次检查，先找出位数尽可能多但不超过限制的数字组合，然后进行排序算法确保数字组合的最大。但我写不出来，，题解的代码需要细细品味。

代码：

```
def f(string):
    if string=='':
        return 0
    else:
        return int(string)
m=int(input())
n=int(input())
l=input().split()
for i in range(n):
    for j in range(n-1-i):
        if l[j] + l[j+1] > l[j+1] + l[j]:
            l[j],l[j+1] = l[j+1],l[j]
weight=[]
for num in l:
    weight.append(len(num))
dp=[['']*(m+1) for _ in range(n+1)]
for k in range(m+1):
    dp[0][k]=''
for q in range(n+1):
    dp[q][0]=''
for i in range(1,n+1):
    for j in range(1,m+1):
        if weight[i-1]>j:
            dp[i][j]=dp[i-1][j]
        else:
                dp[i][j]=str(max(f(dp[i-1][j]),int(l[i-1]+dp[i-1][j-weight[i-1]])))
print(dp[n][m])
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



状态: Accepted

源代码

```
def f(string):
    if string=='':
        return 0
    else:
        return int(string)
```

### 02811: 熄灯问题

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

思路：超时超时超时。以为 brute force 就可以大胆写（天真）又是看代码理解代码学习代码的一题。怎么输入这么多大佬思路还是没有感觉呢，可能不是自己想出来的罢。好像又陷入了奇怪的闭环。

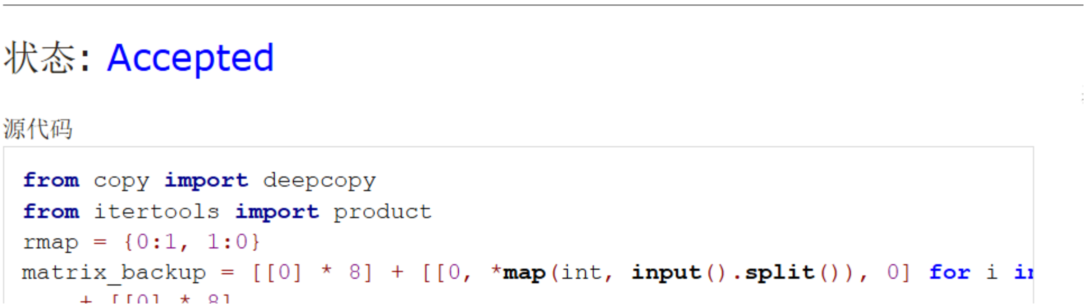
代码：

```
from copy import deepcopy
from itertools import product
rmap = {0:1, 1:0}
```

```python
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 <mark>（至少包含有"Accepted"）</mark>



状态: Accepted

源代码

```python
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
    + [[0] * 8]
```

### 08210: 河中跳房子

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

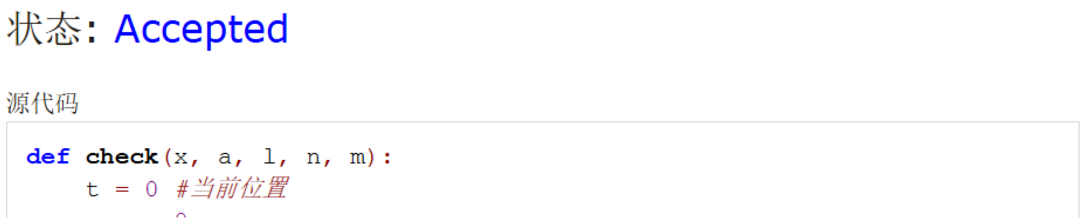
思路：这也是某年的期末题。二分查找题。

代码：

```python
def check(x, a, l, n, m):
    t = 0 #当前位置
    num = 0
    for i in range(n):
        if a[i] - t < x:
            num += 1   #移除
        else:
            t = a[i]   #更新位置
    if l - t < x:
        num += 1   #检查终点距离
    return num <= m   #检查移除点的数量是否小于等于 M

l, n, m = map(int, input().split())
a=[]
for i in range(n):
   a.append(int(input()))
a.sort()
le = 0
ri = l
while le + 1 < ri:
    mid = (le + ri) // 2
    if check(mid, a, l, n, m):
        le = mid   # 更大距离
    else:
        ri = mid   # 减小范围
print(le)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



状态: Accepted

源代码

```
def check(x, a, l, n, m):
    t = 0 #当前位置
```

## 2. 学习总结和收获

复习中，以往年期末题优先，主攻容易和中档题，保持手感；不断完善自己的cheatsheet；复习月考题和自己之前写过的代码，找找思路；复习模板题的做法；模拟考场心态，调整状态。