

# Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Compiled by 同学的姓名、院系

## 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:

常规思路: 把每个硬币三种情况枚举一遍，总24种情况，会写，能接受

但是，为什么用集合做不对啊!!!（怒

代码:

```
for _ in range(int(input())):
    full = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L'}
    fake, heavy, light = full, set(), set()
    for i in range(3):
        a, b, status = input().split()
        a, b = {num for num in a}, {num for num in b}
        if status == 'even':
            fake -= (a|b)
        elif status == 'up':
            fake &= (a|b)
            heavy = heavy|a
            light = light|b
        else:
            fake &= (a|b)
            heavy = heavy|b
            light = light|a
    coin = fake.pop()
    if coin in heavy:
        print(coin + ' is the counterfeit coin and it is heavy.')
    else:
        print(coin + ' is the counterfeit coin and it is light.')
```

代码运行截图 (至少包含有"Accepted")

## #47802164提交状态

状态: Accepted

源代码

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

### 01088: 滑雪

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

思路:

做这道题的心路历程:

不用dp吧? ->不对怎么一直re->哦可以纯dp不dfs->还是要dfs->不要->要->.....

脑子不清楚就是累啊

代码:

```
D = [[-1, 0], [0, -1], [0, 1], [1, 0]]
def dfs(x, y):
    if dp[x][y] > 1:
        return dp[x][y]
    else:
        for ele in D:
            nx, ny = x + ele[0], y + ele[1]
            if 0 <= nx < m and 0 <= ny < n and h[nx][ny] < h[x][y]:
                dp[x][y] = max(dfs(nx, ny) + 1, dp[x][y])
        return dp[x][y]

m, n = map(int, input().split())
h = []
for i in range(m):
    h.append(list(map(int, input().split())))
count = 0
dp = [[1 for _ in range(n)] for _ in range(m)]
for i in range(m):
    for j in range(n):
        count = max(count, dfs(i, j))
print(count)
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
D = [[-1, 0], [0, -1], [0, 1], [1, 0]]
def dfs(x, y):
    if dp[x][y] > 1:
        return dp[x][y]
    else:
        for ele in D:
            nx, ny = x + ele[0], y + ele[1]
            if 0 <= nx < m and 0 <= ny < n and h[nx][ny] < h[x][y]:
                dp[x][y] = max(dfs(nx, ny) + 1, dp[x][y])
        return dp[x][y]

m, n = map(int, input().split())
```

## 25572: 螃蟹采蘑菇

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

思路:

发现了, 纯搜索题只要明确循环和减枝的条件后就会顺利很多,

(一个没用的发现): 把是否能进行下一步循环的条件单独设立成一个函数看上去会好看一点

代码:

```
from collections import deque
D = [[1, 0], [-1, 0], [0, 1], [0, -1]]
def jud(maps, x, y):
    return 0 <= x < n and 0 <= y < n and maps[x][y] != '1'

def bfs(x, y, sta):
    queue, vis = deque([(x, y)]), set((x, y))
    while queue:
        x, y = queue.popleft()
        if maps[x][y] == '9' or maps[x + D[sta][0]][y + D[sta][1]] == '9':
            return 'yes'
        else:
            for i in range(4):
                nx, ny = x + D[i][0], y + D[i][1]
                if jud(maps, nx, ny) and jud(maps, nx + D[sta][0], ny + D[sta][1]) and (nx, ny) not in vis:
                    queue.append((nx, ny))
                    vis.add((nx, ny))
    return 'no'

n = int(input())
maps = []
for _ in range(n):
    maps.append(list(input().split()))
flag = True
for i in range(n):
    for j in range(n):
```

```

if maps[i][j] == '5' and flag:
    if i + 1 < n and maps[i + 1][j] == '5':
        print(bfs(i, j, 0))
    else:
        print(bfs(i, j, 2))
flag = False
break

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

from collections import deque
D = [[1, 0], [-1, 0], [0, 1], [0, -1]]
def jud(maps, x, y):
    return 0 <= x < n and 0 <= y < n and maps[x][y] != '1'

def bfs(x, y, sta):
    queue, vis = deque([(x, y)]), set((x, y))
    while queue:

```

## 27373: 最大整数

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

思路:

dp真的就是拍完脑袋之后一气呵成，乍一看以为是本次最难的题没想到竟然是解决最顺畅的一道题.....

第一步，对各个元素进行估价：第一位越大越好，除此之外位数越短越好。

第二步，确定dp类型：这里我发现可以用类似于割绳子的思路来进行规划，为了防止出现元素重复使用，增加一个列表来存储已经访问过的数

第三步，找到满足条件的位数最长的数进行输出，ac!

代码:

```

m = int(input())
n = int(input())
nums = list(input().split())
nums.sort(key= lambda x:(x[0], len(x)), reverse = True)

dp = ['']*(m + 1)
vis = [[]]*(m + 1)
for i in range(m + 1):
    for j in range(n):
        if i == 0 or (dp[i] != '' and len(nums[j]) + i <= m and j not in vis[i]):
            if dp[i] + nums[j] > dp[i + len(nums[j])]:
                dp[i + len(nums[j])] = dp[i] + nums[j]
                vis[i + len(nums[j])] = vis[i] + [j]

```

```
for i in range(m, -1, -1):
    if dp[i] != '':
        print(dp[i])
        break
```

代码运行截图 (至少包含有"Accepted")

## #47830886提交状态

状态: Accepted

看

源代码

```
m = int(input())
n = int(input())
nums = list(input().split())
nums.sort(key= lambda x: (x[0], len(x)), reverse = True)

dp = [''] * (m + 1)
vis = [[]] * (m + 1)
for i in range(m + 1):
```

## 02811: 熄灯问题

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

思路:

学习了deepcopy和product, 涨知识了 (话说我不是下周就要期末考了嘛怎么还处于涨知识阶段?)

看了题目描述之后有了基本的思路: 后四行的死活取决于第一行的处理方式, 于是枚举第一行的开关, 后五行只要看上方的那个灯开不开就可以, 然后实时进行更改, 如果第五行可以全部关闭就是成功的开关方式, 输出

代码:

```
from copy import deepcopy
from itertools import product
D = [[0, 0], [1, 0], [-1, 0], [0, 1], [0, -1]]
lamps = [[0]*8]
for i in range(5):
    lamps.append([0] + list(map(int, input().split())) + [0])
    lamps.append([0]*8)

for test in product(range(2), repeat=6):
    matrix = deepcopy(lamps)
    triggers = [list(test)]
    for i in range(1, 6):
        for j in range(1, 7):
            if triggers[i - 1][j - 1] != 0:
                for ele in D:
                    di, dj = i + ele[0], j + ele[1]
                    matrix[di][dj] = 1 - matrix[di][dj]
                triggers.append(matrix[i][1:7])
```

```

if matrix[5][1:7] == [0]*6:
    for lines in triggers[:-1]:
        print(' '.join(map(str, lines)))

```

代码运行截图 (至少包含有"Accepted")

## #47831563提交状态

状态: Accepted

基本信息

源代码

```

from copy import deepcopy
from itertools import product
D = [[0, 0], [1, 0], [-1, 0], [0, 1], [0, -1]]
lamps = [[0]*8]
for i in range(5):
    lamps.append([0] + list(map(int, input().split())) + [0])
lamps.append([0]*8)

for test in product(range(2), repeat=6):
    matrix = deepcopy(lamps)
    triggers = [list(test)]
    for i in range(1, 6):

```

题  
提交  
内  
时  
话  
提交时

## 08210: 河中跳房子

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

思路:

幸好给了二分查找的标签, 不然我是绝对想不到做法的.....即便如此在最后逻辑(即全部挪去)中还是参考了一下标答

通过函数在不改变框架的同时进行石头的移去, 根据移去石头数目来调整最小距离

代码:

```

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

def remove(l, m):
    count, d = 0, 0
    for i in range(1, n + 2):
        if stones[i] - d < l:
            count += 1
        else:
            d = stones[i]
    return count > m

left, right = 0, l + 1
while left < right:

```

```
mid = (left + right)//2
if remove(mid, m):
    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())
stones = [0]
for i in range(n):
    stones.append(int(input()))
stones.append(l)

def remove(l, m):
    count, d = 0, 0
    for i in range(1, n + 2):
        if stones[i] - d < l:
            count += 1
```

## 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。

作业越做越崩溃, 好难啊??? 我下周能做出来一道吗?

下周要完蛋了, 能救回来一点是一点吧