Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Complied by <mark>同学的姓名、院系</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：用两个 if 语句分别判断假币为重和假币为轻的情况（比如第一个 if 中，再分在左边，在右边与不在天平上这三种情况讨论）（30min）

代码：

```
n=int(input())
letter=list('ABCDEFGHIJKL')
for _ in range(n):
    cases=[[x for x in input().split()] for _ in range(3)]
    for i in letter:
        if all((i in j[0] and j[2]=='up') or (i in j[1] and j[2]=='down') or (i not in j[0]+j[1] and j[2]=='even')for j in cases):
            print(f'{i} is the counterfeit coin and it is heavy.')
```

```
        if all ((i in j[0] and j[2]=='down') or (i in j[1]
and j[2]=='up') or(i  not in j[0]+j[1] and
j[2]=='even')for j in cases):
            print(f'{i} is the counterfeit coin and it is
light.')
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
def compare(x,y):
    return 1 if x+y<y+x else -1
def insert_sorted(lst,new_elem):
    for i in range(len(lst)):
        if compare(new_elem,lst[i])==-1:
            return lst[:i] + [new_elem]+lst[i:]
    return lst+[new_elem]
m=int(input())
n=int(input())
num=list(map(str,input().split()))
len_num=[len(num[i]) for i in range(n)]
dp=[[[],0] for _ in range(m+1)] for _ in range(n+1)]
for i in range(1,n+1):
    for j in range(1,m+1):
        if j<len_num[i-1]:
            dp[i][j]=dp[i-1][j]
        else:
            num_i1_j=dp[i-1][j][1]
            dp_ij_list=insert_sorted(dp[i-1][j-len_num[i-1]][0],num[i-1
```

基本信息

#:	47899599
题目:	27373
提交人:	24n2400011009
内存:	31716kB
时间:	892ms
语言:	Python3
提交时间:	2024-12-22 17:18:02

01088: 滑雪

```
dp, dfs similar,

http://cs101.openjudge.cn/practice/01088
```

```
思路：一道经典的 dfs 题，memo 用于记录已经计算过的点的最长滑坡长度，初始化为 -1，然后 dfs 深度搜索即可。（1h）
```

```
代码：

import sys

sys.setrecursionlimit(100000)

R, C = map(int, input().split())

area = [list(map(int, input().split())) for _ in range(R)]

memo = [[-1] * C for _ in range(R)]

dx = [-1, 1, 0, 0]

dy = [0, 0, -1, 1]

def dfs(x, y):

    global memo

    if memo[x][y]!= -1:

        return memo[x][y]

    max_length = 1

    for i in range(4):
```

```
        new_x = x + dx[i]
        new_y = y + dy[i]
        if 0 <= new_x < R and 0 <= new_y < C and
area[new_x][new_y] < area[x][y]:
            length = dfs(new_x, new_y) + 1
            max_length = max(max_length, length)
    memo[x][y] = max_length
    return max_length
ans = 0
for i in range(R):
    for j in range(C):
        ans = max(ans, dfs(i, j))
print(ans)
```

```
代码运行截图 ==（至少包含有"Accepted"）==
```

查看 提交 统计 提问

状态: Accepted

源代码

```
import sys
sys.setrecursionlimit(100000)
# 读取输入的行数和列数
R, C = map(int, input().split())
# 存储区域高度信息的二维数组
area = [list(map(int, input().split())) for _ in range(R)]
# 用于记录已经计算过的点的最长滑坡长度，初始化为 -1
memo = [[-1] * C for _ in range(R)]
# 定义四个方向的偏移量，分别对应上下左右
dx = [-1, 1, 0, 0]
dy = [0, 0, -1, 1]
def dfs(x, y):
    global memo
    if memo[x][y]!= -1:
        return memo[x][y]
    max_length = 1
    for i in range(4):
        new_x = x + dx[i]
        new_y = y + dy[i]
        if 0 <= new_x < R and 0 <= new_y < C and area[new_x][new_y] < a
            length = dfs(new_x, new_y) + 1
            max_length = max(max_length, length)
    memo[x][y] = max_length
    return max_length
ans = 0
```

基本信息

#: 47897944
题目: 01088
提交人: 24n2400011009
内存: 4468kB
时间: 48ms
语言: Python3
提交时间: 2024-12-22 16:29:07

25572: 螃蟹采蘑菇

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

思路：用 find_start_end 函数找到螃蟹的初始位置和目标所在的位置，然后再是一个常见的 bfs 模版，最后运行主程序并输出结果。（2h）

代码:

```
from collections import deque
directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]
def find_start_end(n, grid):
    start = []
    end = None
    for i in range(n):
        for j in range(n):
            if grid[i][j] == 5:
                start.append((i, j))
                if j + 1 < n and grid[i][j + 1] == 5:
                    start.append((i, j + 1))
                elif j - 1 >= 0 and grid[i][j - 1] == 5:
                    start.append((i, j - 1))
            elif grid[i][j] == 9:
                end = (i, j)
    return start, end
def bfs(n, grid):
    start, end = find_start_end(n, grid)
    q = deque([start])
    visited = set()
    visited.add(tuple(start))
```

```
    flag = False
    while q:
        if flag:
            break
        current = q.popleft()
        if end in current:
            flag = True
            break
        for dx, dy in directions:
            nx1, ny1 = current[0][0]+dx,
current[0][1]+dy
            nx2, ny2 = current[1][0]+dx,
current[1][1]+dy
            if 0 <= nx1 < n and 0 <= ny1 < n and 0 <= nx2
< n and 0 <= ny2 < n:
                if grid[nx1][ny1] != 1 and
grid[nx2][ny2] != 1:
                    next_state = [(nx1, ny1), (nx2, ny2)]
                    if tuple(next_state) not in visited:
                        visited.add(tuple(next_state))
                        q.append(next_state)
    return flag
```

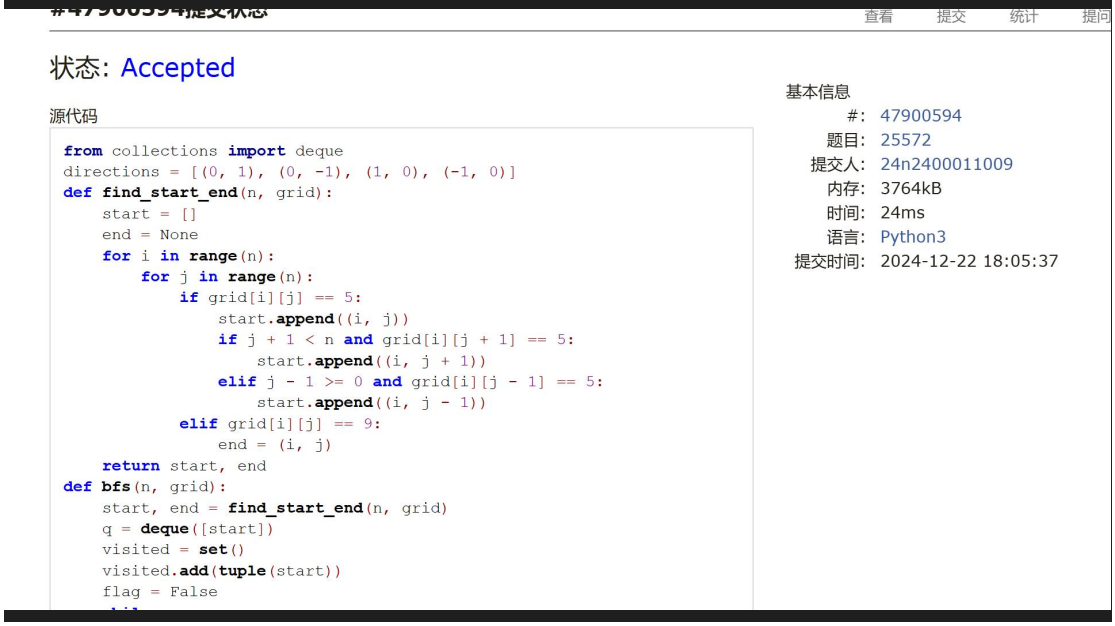
```
n=int(input())

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

result = bfs(n, grid)

print("yes" if result else "no")
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



状态: Accepted

源代码

```
from collections import deque
directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]
def find_start_end(n, grid):
    start = []
    end = None
    for i in range(n):
        for j in range(n):
            if grid[i][j] == 5:
                start.append((i, j))
                if j + 1 < n and grid[i][j + 1] == 5:
                    start.append((i, j + 1))
                elif j - 1 >= 0 and grid[i][j - 1] == 5:
                    start.append((i, j - 1))
            elif grid[i][j] == 9:
                end = (i, j)
    return start, end
def bfs(n, grid):
    start, end = find_start_end(n, grid)
    q = deque([start])
    visited = set()
    visited.add(tuple(start))
    flag = False
```

查看 提交 统计 提问

基本信息

#: 47900594
题目: 25572
提交人: 24n2400011009
内存: 3764kB
时间: 24ms
语言: Python3
提交时间: 2024-12-22 18:05:37

27373: 最大整数

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

思路：与背包问题有相似之处，每次计算加入一个新数字后的最大整数，与不加入这个数字时的不超过 m 位的最大整数相比较。（3h）

代码：

```
def compare(x,y):
    return 1 if x+y<y+x else -1
def insert_sorted(lst,new_elem):
    for i in range(len(lst)):
        if compare(new_elem,lst[i])==-1:
            return lst[:i] + [new_elem]+lst[i:]
    return lst+[new_elem]
m=int(input())
n=int(input())
num=list(map(str,input().split()))
len_num=[len(num[i]) for i in range(n)]
dp=[[[],0] for _ in range(m+1)] for _ in range(n+1)]
for i in range(1,n+1):
    for j in range(1,m+1):
        if j<len_num[i-1]:
            dp[i][j]=dp[i-1][j]
```

```
        else:
            num_i1_j=dp[i-1][j][1]
            dp_ij_list=insert_sorted(dp[i-1][j-len_num
[i-1]][0],num[i-1])
            num_ij_str=''.join(dp_ij_list)
            num_ij=int(num_ij_str) if num_ij_str else 0
            if num_ij>=num_i1_j:
                dp[i][j][0]=dp_ij_list
                dp[i][j][1]=num_ij
            else:
                dp[i][j]=dp[i-1][j]
result=dp[-1][-1][1]
print(result if result else '0')
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
def compare(x,y):
    return 1 if x+y<y+x else -1
def insert_sorted(lst,new_elem):
    for i in range(len(lst)):
        if compare(new_elem,lst[i])==-1:
            return lst[:i] + [new_elem]+lst[i:]
    return lst+[new_elem]
m=int(input())
n=int(input())
num=list(map(str,input().split()))
len_num=[len(num[i]) for i in range(n)]
dp=[[[],0] for _ in range(m+1)] for _ in range(n+1)]
for i in range(1,n+1):
    for j in range(1,m+1):
        if j<len_num[i-1]:
            dp[i][j]=dp[i-1][j]
        else:
            num_i1_j=dp[i-1][j][1]
            dp_ij_list=insert_sorted(dp[i-1][j-len_num[i-1]][0],num[i-1
```

基本信息

#: 47899599
题目: 27373
提交人: 24n2400011009
内存: 31716kB
时间: 892ms
语言: Python3
提交时间: 2024-12-22 17:18:02

02811: 熄灯问题

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

思路：一是枚举第一行的情况，二是从第一行往后进行递推，基于上一行灯的状态确定下一行的情况。三是通过位运算进行状态判断与记录。（耗时:float('inf')）

代码：

```
dx, dy=[0,0,-1,1,0],[0,-1,0,0,1]
```

```
def press(light,x,y):
    for i in range(5):
        nx, ny=x+ dx[i],y+ dy[i]
        if 0<= nx<5 and 0<= ny<6:
            light[nx][ny]^=1
def doit():
    for first_row in range(64):
        s=[row[:] for row in light]
        solution=[[0]*6 for _ in range(5)]
        for j in range(6):
            if (first_row >> j)&1:
                solution[0][j]=1
                press(s,0,j)
        for i in range(1,5):
            for j in range(6):
                if s[i-1][j]==1:
                    solution[i][j]=1
                    press(s,i,j)
        if all(s[4][j]==0 for j in range(6)):
            for i in solution:
                print(*i)
```

```
light=[[int(i) for i in input(). split()] for _ in
range(5)]
doit()
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47900266提交状态

查看 提交 统计 提

状态: Accepted

源代码

```
dx, dy=[0,0,-1,1,0],[0,-1,0,0,1]
def press(light,x,y):
    for i in range(5):
        nx, ny=x+ dx[i],y+ dy[i]
        if 0<= nx<5 and 0<= ny<6:
            light[nx][ny]^=1
def doit():
    for first_row in range(64):
        s=[row[:] for row in light]
        solution=[[0]*6 for _ in range(5)]
        for j in range(6):
            if (first_row >> j)&1:
                solution[0][j]=1
                press(s,0,j)
        for i in range(1,5):
            for j in range(6):
                if s[i-1][j]==1:
                    solution[i][j]=1
                    press(s,i,j)
        if all(s[4][j]==0 for j in range(6)):
            for i in solution:
                print(*i)
```

基本信息

#: 47900266
题目: 02811
提交人: 24n2400011009
内存: 3660kB
时间: 24ms
语言: Python3
提交时间: 2024-12-22 17:46:37

```
### 08210: 河中跳房子
```

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

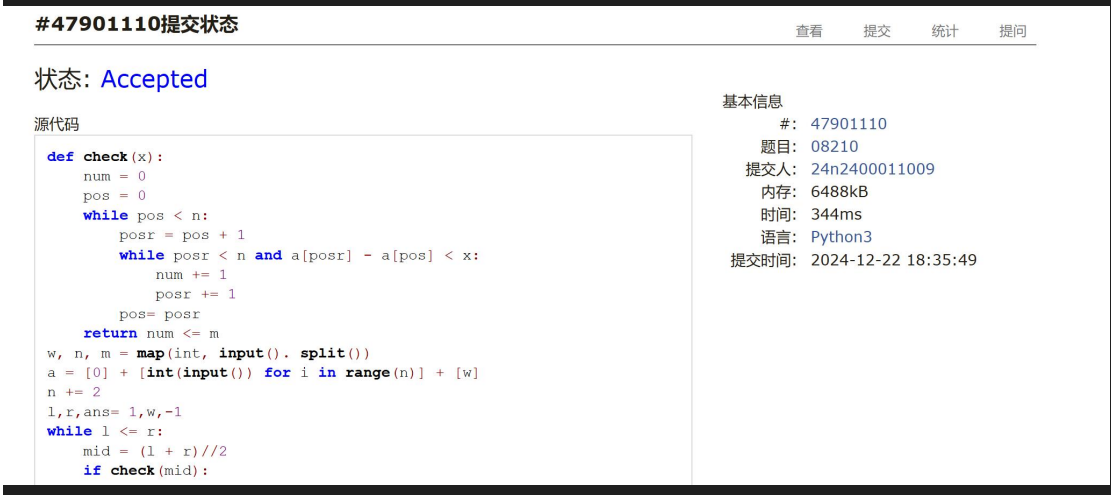
思路:主要利用二分查找来调整区间,通过移除一定数量(最多 m 个)的元素，使得剩余元素之间的最小间隔尽可能大，最终找到这个最大的最小间隔值并输出。（3h）

代码:

```
def check(x):
    num = 0
    pos = 0
    while pos < n:
        posr = pos + 1
        while posr < n and a[posr] - a[pos] < x:
            num += 1
            posr += 1
        pos= posr
    return num <= m
w, n, m = map(int, input(). split())
a = [0] + [int(input()) for i in range(n)] + [w]
n += 2
l,r,ans= 1,w,-1
while l <= r:
    mid = (l + r)//2
```

```
    if check(mid):
        ans, l = mid, mid + 1
    else:
        r = mid - 1
print(ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



#47901110提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def check(x):
    num = 0
    pos = 0
    while pos < n:
        posr = pos + 1
        while posr < n and a[posr] - a[pos] < x:
            num += 1
            posr += 1
        pos= posr
    return num <= m
w, n, m = map(int, input(). split())
a = [0] + [int(input()) for i in range(n)] + [w]
n += 2
l,r,ans= 1,w,-1
while l <= r:
    mid = (l + r)//2
    if check(mid):
```

基本信息

#:	47901110
题目:	08210
提交人:	24n2400011009
内存:	6488kB
时间:	344ms
语言:	Python3
提交时间:	2024-12-22 18:35:49

2. 学习总结和收获

<mark>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。</mark>

好难好难，被最后一次作业直接创飞，红温了，机考只能指望 cheating paper 了，跳大神啊跳大神！！！

球球 E 难度简单一点啊啊啊呜呜呜呜呜呜