Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Complied by 徐贤天，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：

一些集合的交与并与补（居然过了）

代码：

```
n = int(input())
coins = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L'}
for _ in range(n):
    true_coins = set()
    light_set = set()
    heavy_set = set()
    for _ in range(3):
        a, b, c = input().split()
        if c == 'even':
            true_coins = true_coins.union(set(a), set(b))
        elif c == 'up':
            light_set = light_set.union(set(b))
            heavy_set = heavy_set.union(set(a))
        elif c == 'down':
            light_set = light_set.union(set(a))
            heavy_set = heavy_set.union(set(b))
    wrong_in_it1 = light_set.difference(heavy_set)
    wrong_in_it1 = wrong_in_it1.union(heavy_set.difference((light_set)))
    wrong_in_it2 = coins.difference(true_coins)
    wrong = wrong_in_it2.intersection(wrong_in_it1)
    for i in wrong:
        if i in light_set:
            print('{} is the counterfeit coin and it is {}. '.format(i, 'light'))
            break
```

```
        elif i in heavy_set:
            print('{} is the counterfeit coin and it is {}. '.format(i, 'heavy'))
            break
```

代码运行截图 （至少包含有"Accepted"）

#47837074提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n = int(input())
coins = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L'}
for _ in range(n):
    true_coins = set()
    light_set = set()
    heavy_set = set()
    for _ in range(3):
        a, b, c = input().split()
        if c == 'even':
            true_coins = true_coins.union(set(a), set(b))
        elif c == 'up':
            light_set = light_set.union(set(b))
            heavy_set = heavy_set.union(set(a))
        elif c == 'down':
            light_set = light_set.union(set(a))
            heavy_set = heavy_set.union(set(b))
    wrong_in_it1 = light_set.difference(heavy_set)
    wrong_in_it1 = wrong_in_it1.union(heavy_set.difference((light_set))
    wrong_in_it2 = coins.difference(true_coins)
    wrong = wrong_in_it2.intersection(wrong_in_it1)
    for i in wrong:
        if i in light_set:
            print('{} is the counterfeit coin and it is {}. '.format(i, 'light')
            break
        elif i in heavy_set:
            print('{} is the counterfeit coin and it is {}. '.format(i, 'heavy')
            break
```

基本信息

#: 47837074
题目: 02692
提交人: 24n2400011033
内存: 3600kB
时间: 23ms
语言: Python3
提交时间: 2024-12-19 14:27:48

English 帮助 关于

01088: 滑雪

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

思路:

用dp数组实现记忆化搜索

代码:

```
r, c = map(int, input().split())
matrix = [[int(x) for x in input().split()] for _ in range(r)]
directions = [(1, 0), (-1, 0), (0, 1), (0 ,-1)]
ans = 0
dp = [[0] * c for _ in range(r)]

def dfs(x, y):
    max_length = 0
    for dx, dy in directions:
        nx, ny = x + dx, y + dy
        if 0 <= nx < r and 0 <= ny < c and matrix[nx][ny] < matrix[x][y]:
            if dp[nx][ny] > 0:
                max_length = max(max_length, dp[nx][ny] + 1)
```

```
            else:
                max_length = max(max_length, dfs(nx, ny) + 1)
    dp[x][y] = max_length
    return max_length

for i in range(r):
    for j in range(c):
        ans = max(ans, dfs(i, j))
print(ans+1)
```

代码运行截图 ==（至少包含有"Accepted"）==

25572: 螃蟹采蘑菇

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

思路：

相当于正常的bfs在后面拖了个尾巴，需要加入额外的判断条件

代码：

```
from collections import deque
n = int(input())
matrix = [[int(x) for x in input().split()] for _ in range(n)]
directions = [(1, 0), (-1, 0), (0, -1), (0, 1)]

def find_start():
    for i in range(n):
        for j in range(n):
            if matrix[i][j] == 5:
                return i, j
s_x, s_y = find_start()
```

```
def find_other():
    for dx, dy in directions:
        nx, ny = s_x + dx, s_y + dy
        if 0 <= nx < n and 0 <= ny < n and matrix[nx][ny] == 5:
            o_x, o_y = nx, ny
            return o_x, o_y
o_x, o_y = find_other()
d_o_x, d_o_y = o_x - s_x, o_y - s_y

def bfs():
    q = deque([(s_x, s_y)])
    inq = {(s_x, s_y)}
    while q:
        x, y = q.popleft()
        if matrix[x][y] == 9 or matrix[x + d_o_x][y + d_o_y] == 9:
            return 'yes'
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < n and 0 <= ny < n and (nx, ny) not in inq:
                if 0 <= nx + d_o_x < n and 0 <= ny + d_o_y < n:
                    if matrix[nx][ny] != 1 and matrix[nx + d_o_x][ny + d_o_y] !=
1:
                        q.append((nx, ny))
                        inq.add((nx, ny))
    return 'no'

print(bfs())
```

代码运行截图 （至少包含有"Accepted"）

#47838376提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque
n = int(input())
matrix = [[int(x) for x in input().split()] for _ in range(n)]
directions = [(1, 0), (-1, 0), (0, -1), (0, 1)]

def find_start():
    for i in range(n):
        for j in range(n):
            if matrix[i][j] == 5:
                return i, j
s_x, s_y = find_start()

def find_other():
    for dx, dy in directions:
        nx, ny = s_x + dx, s_y + dy
        if 0 <= nx < n and 0 <= ny < n and matrix[nx][ny] == 5:
            o_x, o_y = nx, ny
            return o_x, o_y
o_x, o_y = find_other()
d_o_x, d_o_y = o_x - s_x, o_y - s_y

def bfs():
    q = deque([(s_x, s_y)])
    inq = {(s_x, s_y)}
    while q:
        x, y = q.popleft()
        if matrix[x][y] == 9 or matrix[x + d_o_x][y + d_o_y] == 9:
            return 'yes'
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < n and 0 <= ny < n and (nx, ny) not in inq:
                if 0 <= nx + d_o_x < n and 0 <= ny + d_o_y < n:
                    if matrix[nx][ny] != 1 and matrix[nx + d_o_x][ny +
                        q.append((nx, ny))
                        inq.add((nx, ny))
    return 'no'

print(bfs())
```

基本信息

#:	47838376
题目:	25572
提交人:	24n2400011033
内存:	3744kB
时间:	22ms
语言:	Python3
提交时间:	2024-12-19 15:25:06

English 帮助 关于

27373: 最大整数

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

思路:

刷新了最无语的错误：“ 和 ‘’ 没有区分，多了一个空格导致三次RE🤯

代码:

```
m = int(input())
n = int(input())
nums = input().split()

def translate_to_int(a):
    if a == '':
        return 0
    else:
        return int(a)

#冒泡排序，使得任意前后两个数字顺序相加后均大于逆序相加
for i in range(n):
    for j in range(n-i-1):
        if nums[j] + nums[j+1] < nums[j+1] + nums[j]:
```

```
            nums[j], nums[j+1] = nums[j+1], nums[j]

dp = [[''] * (m+1) for _ in range(n)]
num1 = nums[0]
for i in range(m+1):
    if i >= len(num1):
        dp[0][i] = num1

for i in range(1, n):
    length = len(nums[i])
    for j in range(1, m+1):
        if j < length:
            dp[i][j] = dp[i-1][j]
        else:
            dp[i][j] = str(max(translate_to_int(dp[i-1][j]), int(dp[i-1][j-
length] + nums[i])))

print(dp[n-1][m])
```

代码运行截图 （至少包含有"Accepted"）

#47851034提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
m = int(input())
n = int(input())
nums = input().split()

def translate_to_int(a):
    if a == '':
        return 0
    else:
        return int(a)

#冒泡排序，使得任意前后两个数字顺序相加后均大于逆序相加
for i in range(n):
    for j in range(n-i-1):
        if nums[j] + nums[j+1] < nums[j+1] + nums[j]:
            nums[j], nums[j+1] = nums[j+1], nums[j]

dp = [[''] * (m+1) for _ in range(n)]
num1 = nums[0]
for i in range(m+1):
    if i >= len(num1):
        dp[0][i] = num1

for i in range(1, n):
    length = len(nums[i])
    for j in range(1, m+1):
        if j < length:
            dp[i][j] = dp[i-1][j]
        else:
            dp[i][j] = str(max(translate_to_int(dp[i-1][j]), int(dp[i-1
print(dp[n-1][m])
```

基本信息

#:	47851034
题目:	27373
提交人:	24n2400011033
内存:	31252kB
时间:	585ms
语言:	Python3
提交时间:	2024-12-19 23:19:01

English 帮助 关于

02811: 熄灯问题

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

思路：

感觉有点递推的味道，最重要的是想到第一行先枚举出各种组合可能，接下来熄灯的方法便固定了

代码：

```
import copy
matrix = [[int(x) for x in input().split()] for _ in range(5)]
directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

def press(temp):
    grid = copy.deepcopy(matrix)

    def calculate(temp, i):
        for j in range(6):
            if temp[j] == 1:
                grid[i][j] = grid[i][j] ^ 1
                for dx, dy in directions:
                    nx, ny = i + dx, j + dy
                    if 0 <= nx < 5 and 0 <= ny < 6:
                        grid[nx][ny] = grid[nx][ny] ^ 1

    ans = [temp]
    calculate(temp, 0)
    for i in range(4):
        temp = grid[i]
        ans.append(temp[:])
        calculate(temp, i+1)
    if grid[4] == [0]*6:
        for x in ans:
            print(*x)

def permutation(temp):
    if len(temp) == 6:
        press(temp)
        return
    permutation(temp + [0])
    permutation(temp + [1])

permutation([])
```

代码运行截图 （至少包含有"Accepted"）

#47864406提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
import copy
matrix = [[int(x) for x in input().split()] for _ in range(5)]
directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

def press(temp):
    grid = copy.deepcopy(matrix)

    def calculate(temp, i):
        for j in range(6):
            if temp[j] == 1:
                grid[i][j] = grid[i][j] ^ 1
                for dx, dy in directions:
                    nx, ny = i + dx, j + dy
                    if 0 <= nx < 5 and 0 <= ny < 6:
                        grid[nx][ny] = grid[nx][ny] ^ 1

    ans = [temp]
    calculate(temp, 0)
    for i in range(4):
        temp = grid[i]
        ans.append(temp[:])
        calculate(temp, i+1)
    if grid[4] == [0]*6:
        for x in ans:
            print(*x)

def permutation(temp):
    if len(temp) == 6:
        press(temp)
        return
    permutation(temp + [0])
    permutation(temp + [1])

permutation([])
```

基本信息

#: 47864406
题目: 02811
提交人: 24n2400011033
内存: 3920kB
时间: 26ms
语言: Python3
提交时间: 2024-12-20 18:43:14

08210: 河中跳房子

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

思路:

感觉二分法最重要的是明确自己代码里的区间是左闭右闭区间或是左闭右开区间，各种条件都按照这个区间的定义来写就不容易出错

代码:

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

def the_num_of_removed_rocks(mid):
    current_position = 0
    cnt = 0
    for i in range(1, n+2):
        if positions[i] - current_position >= mid:
            current_position = positions[i]
        else:
            cnt += 1
    return cnt
```

```python
#针对最短跳跃距离二分
l = 0
r = L
ans = 0
while l <= r:
    mid = (l+r)//2
    if the_num_of_removed_rocks(mid) > m:
        r = mid-1
    elif the_num_of_removed_rocks(mid) == m:
        ans = mid
        l = mid+1
    elif the_num_of_removed_rocks(mid) < m:
        l = mid+1

print(ans)
```

代码运行截图 （至少包含有"Accepted"）

#47885214提交状态

查看 提交 统计 提问

状态: Accepted

源代码

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

def the_num_of_removed_rocks(mid):
    current_position = 0
    cnt = 0
    for i in range(1, n+2):
        if positions[i] - current_position >= mid:
            current_position = positions[i]
        else:
            cnt += 1
    return cnt

#针对最短跳跃距离二分
l = 0
r = L
ans = 0
while l <= r:
    mid = (l+r)//2
    if the_num_of_removed_rocks(mid) > m:
        r = mid-1
    elif the_num_of_removed_rocks(mid) == m:
        ans = mid
        l = mid+1
    elif the_num_of_removed_rocks(mid) < m:
        l = mid+1

print(ans)
```

基本信息

#: 47885214
题目: 08210
提交人: 24n2400011033
内存: 5580kB
时间: 367ms
语言: Python3
提交时间: 2024-12-21 21:03:13

English 帮助 关于

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。

感觉做题逐渐得心应手起来（虽然速度还不够快……）

这次学到了一些东西，比如假币问题里关于集合的一些语法，熄灯问题的巧妙方法，以及二分法里需要注意的细节

还有需要注意 ' '中间的空格😳

期末加油!