Assignment #A: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Complied by 佟永鑫 元培学院

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

LuoguP1255 数楼梯

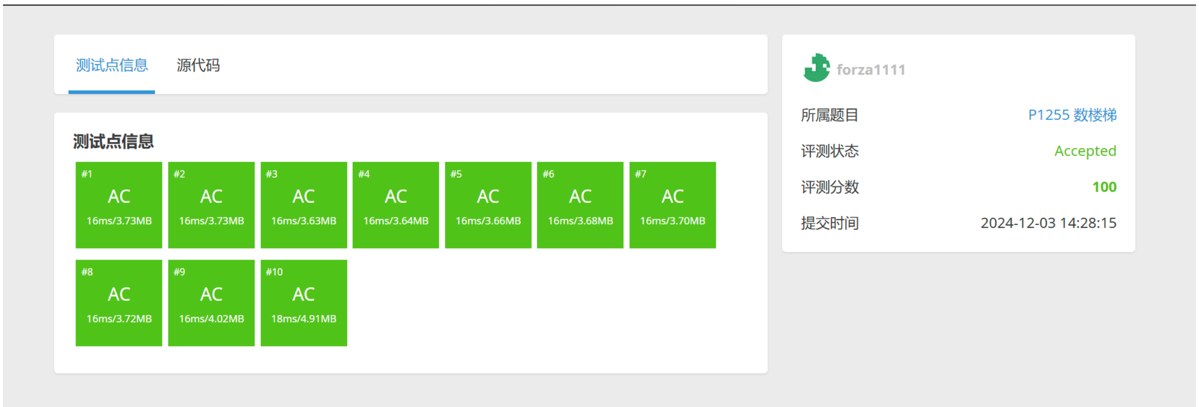
dp, bfs, https://www.luogu.com.cn/problem/P1255

思路：

代码：

```
def count_steps(n):
    if n == 1:
        return 1
    dp = [0] * (n + 1)
    dp[0] = 1
    dp[1] = 1
    for i in range(2, n + 1):
        dp[i] = dp[i - 1] + dp[i - 2]
    return dp[n]
n = int(input())
print(count_steps(n))
```

代码运行截图 （至少包含有"Accepted"）



测试点信息

#1 AC 16ms/3.73MB	#2 AC 16ms/3.73MB	#3 AC 16ms/3.63MB	#4 AC 16ms/3.64MB	#5 AC 16ms/3.66MB	#6 AC 16ms/3.68MB	#7 AC 16ms/3.70MB
#8 AC 16ms/3.72MB	#9 AC 16ms/4.02MB	#10 AC 18ms/4.91MB				

forza1111

所属题目	P1255 数楼梯
评测状态	Accepted
评测分数	100
提交时间	2024-12-03 14:28:15

27528: 跳台阶

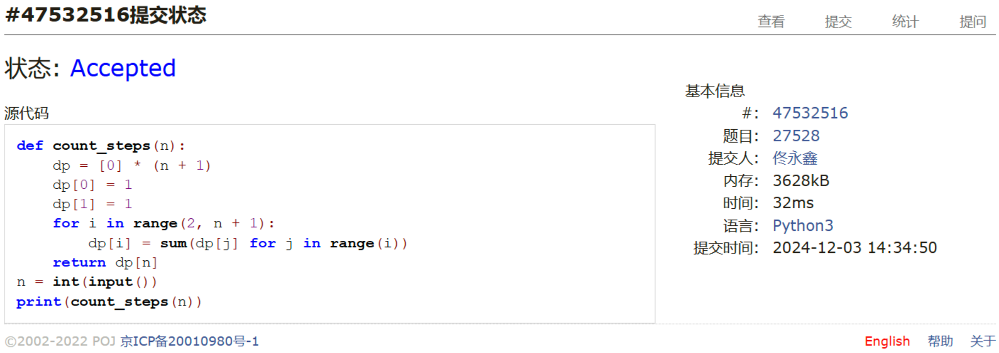
dp, http://cs101.openjudge.cn/practice/27528/

思路：

代码：

```python
def count_steps(n):
    dp = [0] * (n + 1)
    dp[0] = 1
    dp[1] = 1
    for i in range(2, n + 1):
        dp[i] = sum(dp[j] for j in range(i))
    return dp[n]
n = int(input())
print(count_steps(n))
```

代码运行截图 ==（至少包含有"Accepted"）==



#47532516提交状态

状态: Accepted

源代码

```python
def count_steps(n):
    dp = [0] * (n + 1)
    dp[0] = 1
    dp[1] = 1
    for i in range(2, n + 1):
        dp[i] = sum(dp[j] for j in range(i))
    return dp[n]
n = int(input())
print(count_steps(n))
```

基本信息

#:	47532516
题目:	27528
提交人:	佟永鑫
内存:	3628kB
时间:	32ms
语言:	Python3
提交时间:	2024-12-03 14:34:50

©2002-2022 POJ 京ICP备20010980号-1

474D. Flowers

dp, https://codeforces.com/problemset/problem/474/D

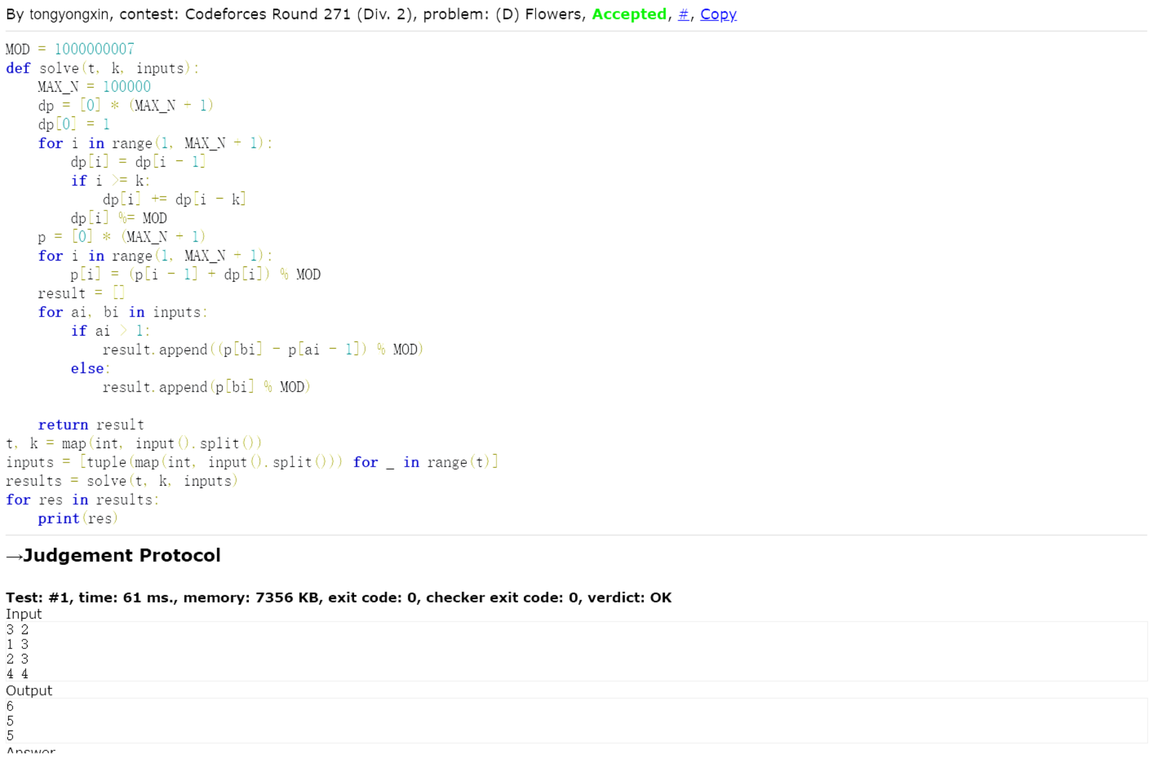
思路：

代码：

```
MOD = 1000000007
def solve(t, k, inputs):
    MAX_N = 100000
    dp = [0] * (MAX_N + 1)
    dp[0] = 1
    for i in range(1, MAX_N + 1):
        dp[i] = dp[i - 1]
        if i >= k:
            dp[i] += dp[i - k]
        dp[i] %= MOD
    p = [0] * (MAX_N + 1)
    for i in range(1, MAX_N + 1):
        p[i] = (p[i - 1] + dp[i]) % MOD
    result = []
    for ai, bi in inputs:
        if ai > 1:
            result.append((p[bi] - p[ai - 1]) % MOD)
        else:
            result.append(p[bi] % MOD)

    return result
t, k = map(int, input().split())
inputs = [tuple(map(int, input().split())) for _ in range(t)]
results = solve(t, k, inputs)
for res in results:
    print(res)
```

代码运行截图 （至少包含有"Accepted"）



By tongyongxin, contest: Codeforces Round 271 (Div. 2), problem: (D) Flowers, Accepted, #, Copy

```
MOD = 1000000007
def solve(t, k, inputs):
    MAX_N = 100000
    dp = [0] * (MAX_N + 1)
    dp[0] = 1
    for i in range(1, MAX_N + 1):
        dp[i] = dp[i - 1]
        if i >= k:
            dp[i] += dp[i - k]
        dp[i] %= MOD
    p = [0] * (MAX_N + 1)
    for i in range(1, MAX_N + 1):
        p[i] = (p[i - 1] + dp[i]) % MOD
    result = []
    for ai, bi in inputs:
        if ai > 1:
            result.append((p[bi] - p[ai - 1]) % MOD)
        else:
            result.append(p[bi] % MOD)

    return result
t, k = map(int, input().split())
inputs = [tuple(map(int, input().split())) for _ in range(t)]
results = solve(t, k, inputs)
for res in results:
    print(res)
```

→Judgement Protocol

Test: #1, time: 61 ms., memory: 7356 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

```
3 2
1 3
2 3
4 4
```

Output

```
6
5
5
```

Answer

LeetCode5.最长回文子串

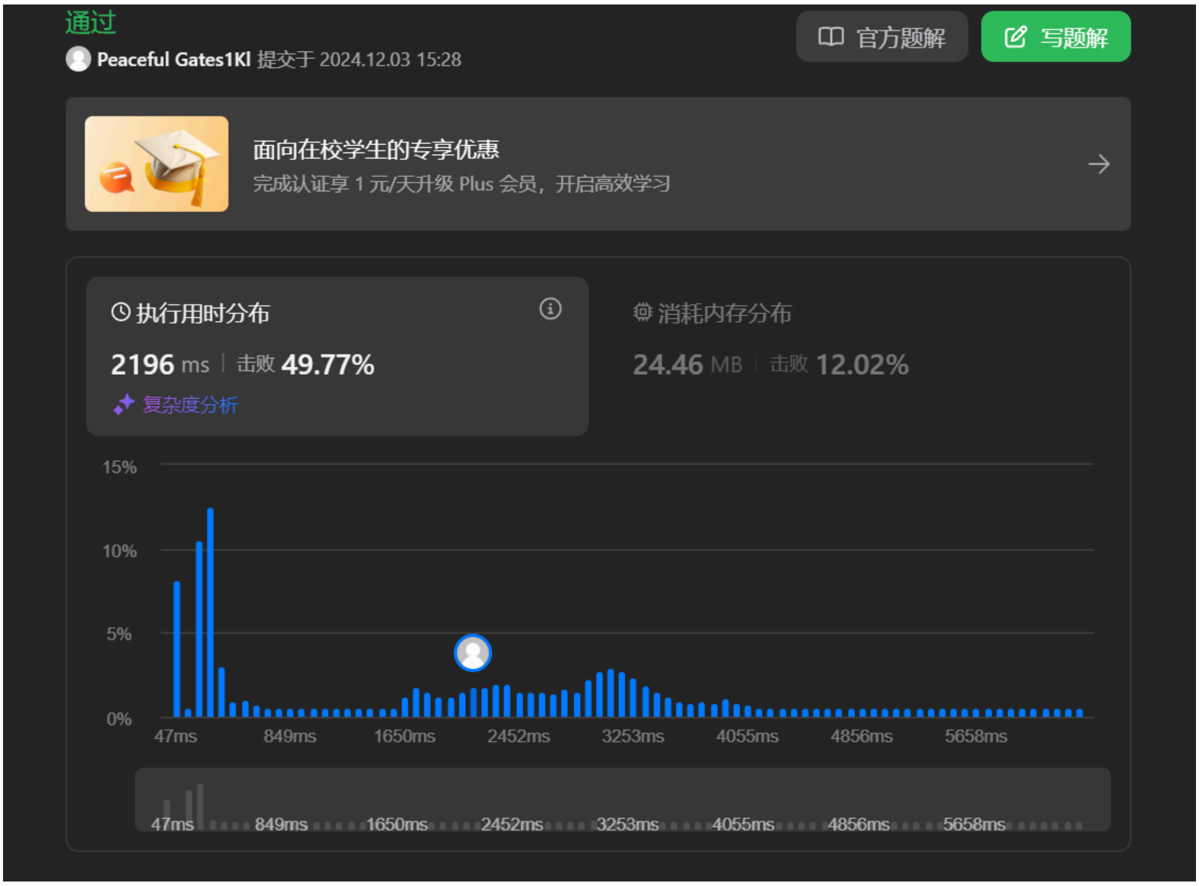
dp, two pointers, string, https://leetcode.cn/problems/longest-palindromic-substring/

思路：

代码：

```
class Solution:
    def longestPalindrome(self, s: str) -> str:
        n = len(s)
        dp = [[False] * n for _ in range(n)]
        start = 0
        max_length = 1
        for i in range(n):
            dp[i][i] = True
        for i in range(n - 1):
            if s[i] == s[i + 1]:
                dp[i][i + 1] = True
                start = i
                max_length = 2
        for length in range(3, n + 1):
            for i in range(n - length + 1):
                j = i + length - 1
                if s[i] == s[j] and dp[i + 1][j - 1]:
                    dp[i][j] = True
                    if length > max_length:
                        start = i
                        max_length = length
        return s[start:start + max_length]
```

代码运行截图 （至少包含有"Accepted"）



通过

Peaceful Gates1Kl 提交于 2024.12.03 15:28

执行用时分布

2196 ms | 击败 49.77%

复杂度分析

消耗内存分布

24.46 MB | 击败 12.02%

12029: 水淹七军

bfs, dfs, http://cs101.openjudge.cn/practice/12029/

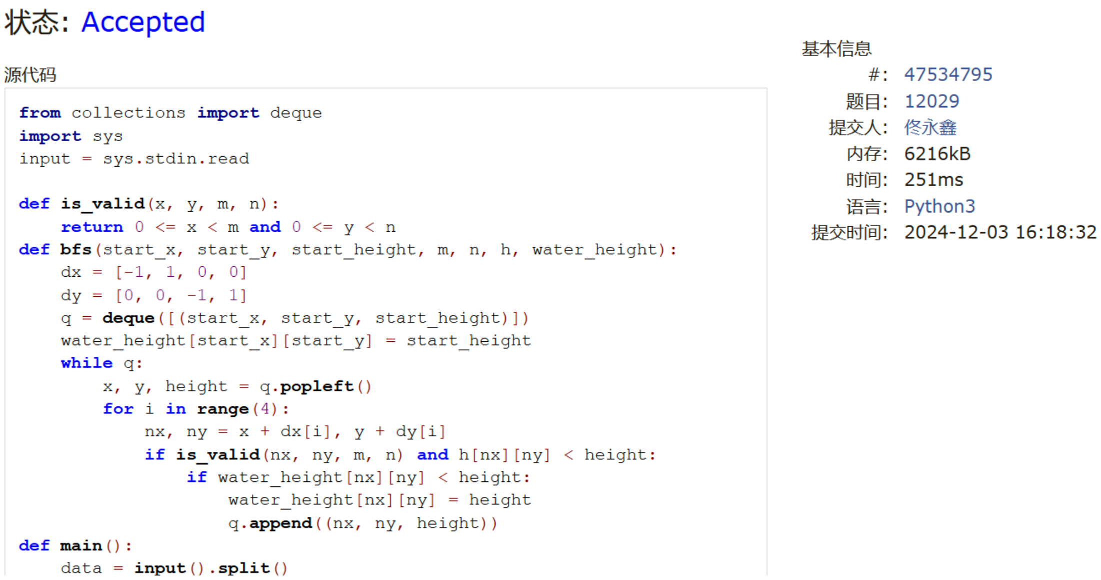
思路：

代码：

```
from collections import deque
import sys
input = sys.stdin.read

def is_valid(x, y, m, n):
    return 0 <= x < m and 0 <= y < n
def bfs(start_x, start_y, start_height, m, n, h, water_height):
    dx = [-1, 1, 0, 0]
    dy = [0, 0, -1, 1]
    q = deque([(start_x, start_y, start_height)])
    water_height[start_x][start_y] = start_height
    while q:
        x, y, height = q.popleft()
        for i in range(4):
            nx, ny = x + dx[i], y + dy[i]
            if is_valid(nx, ny, m, n) and h[nx][ny] < height:
                if water_height[nx][ny] < height:
                    water_height[nx][ny] = height
                    q.append((nx, ny, height))
def main():
    data = input().split()
    idx = 0
    k = int(data[idx])
    idx += 1
    results = []
    for _ in range(k):
        m, n = map(int, data[idx:idx + 2])
        idx += 2
        h = []
        for i in range(m):
            h.append(list(map(int, data[idx:idx + n])))
            idx += n
        water_height = [[0] * n for _ in range(m)]
        i, j = map(int, data[idx:idx + 2])
        idx += 2
        i, j = i - 1, j - 1
        p = int(data[idx])
        idx += 1
        for _ in range(p):
            x, y = map(int, data[idx:idx + 2])
            idx += 2
            x, y = x - 1, y - 1
            if h[x][y] <= h[i][j]:
                continue
            bfs(x, y, h[x][y], m, n, h, water_height)
        results.append("Yes" if water_height[i][j] > 0 else "No")
    sys.stdout.write("\n".join(results) + "\n")
if __name__ == "__main__":
    main()
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
from collections import deque
import sys
input = sys.stdin.read

def is_valid(x, y, m, n):
    return 0 <= x < m and 0 <= y < n
def bfs(start_x, start_y, start_height, m, n, h, water_height):
    dx = [-1, 1, 0, 0]
    dy = [0, 0, -1, 1]
    q = deque([(start_x, start_y, start_height)])
    water_height[start_x][start_y] = start_height
    while q:
        x, y, height = q.popleft()
        for i in range(4):
            nx, ny = x + dx[i], y + dy[i]
            if is_valid(nx, ny, m, n) and h[nx][ny] < height:
                if water_height[nx][ny] < height:
                    water_height[nx][ny] = height
                    q.append((nx, ny, height))
def main():
    data = input().split()
```

基本信息

#:	47534795
题目:	12029
提交人:	佟永鑫
内存:	6216kB
时间:	251ms
语言:	Python3
提交时间:	2024-12-03 16:18:32

02802: 小游戏

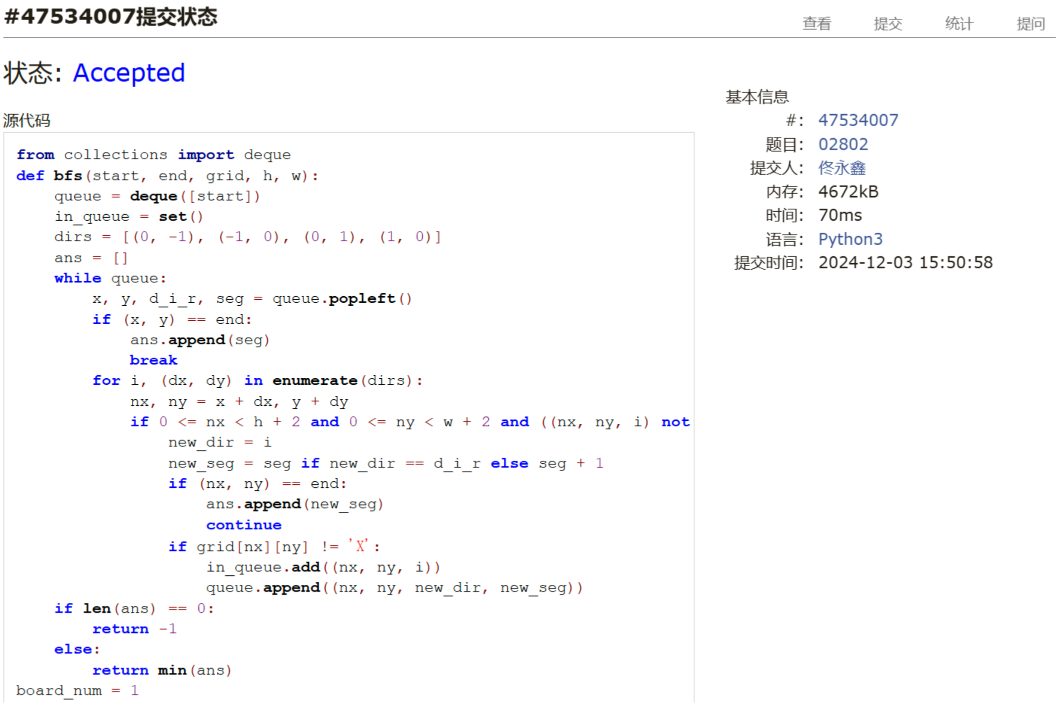
bfs, http://cs101.openjudge.cn/practice/02802/

思路：

代码：

```
from collections import deque
def bfs(start, end, grid, h, w):
    queue = deque([start])
    in_queue = set()
    dirs = [(0, -1), (-1, 0), (0, 1), (1, 0)]
    ans = []
    while queue:
        x, y, d_i_r, seg = queue.popleft()
        if (x, y) == end:
            ans.append(seg)
            break
        for i, (dx, dy) in enumerate(dirs):
            nx, ny = x + dx, y + dy
            if 0 <= nx < h + 2 and 0 <= ny < w + 2 and ((nx, ny, i) not in
in_queue):
                new_dir = i
                new_seg = seg if new_dir == d_i_r else seg + 1
                if (nx, ny) == end:
                    ans.append(new_seg)
                    continue
                if grid[nx][ny] != 'X':
                    in_queue.add((nx, ny, i))
                    queue.append((nx, ny, new_dir, new_seg))
    if len(ans) == 0:
        return -1
    else:
        return min(ans)
board_num = 1
while True:
    w, h = map(int, input().split())
    if w == h == 0:
        break
    grid = [' ' * (w + 2)] + [' ' + input() + ' ' for _ in range(h)] + [' ' * (w
+2)]
    print(f"Board #{board_num}:")
    pair_num = 1
    while True:
        y1, x1, y2, x2 = map(int, input().split())
        if x1 == y1 == x2 == y2 == 0:
            break
        start = (x1, y1, -1, 0)
        end = (x2, y2)
        seg = bfs(start, end, grid, h, w)
        if seg == -1:
            print(f"Pair {pair_num}: impossible.")
        else:
            print(f"Pair {pair_num}: {seg} segments.")
        pair_num += 1
    print()
    board_num += 1
```

代码运行截图 （至少包含有"Accepted"）



#47534007提交状态

状态: Accepted

源代码

```
from collections import deque
def bfs(start, end, grid, h, w):
    queue = deque([start])
    in_queue = set()
    dirs = [(0, -1), (-1, 0), (0, 1), (1, 0)]
    ans = []
    while queue:
        x, y, d_i_r, seg = queue.popleft()
        if (x, y) == end:
            ans.append(seg)
            break
        for i, (dx, dy) in enumerate(dirs):
            nx, ny = x + dx, y + dy
            if 0 <= nx < h + 2 and 0 <= ny < w + 2 and ((nx, ny, i) not
                new_dir = i
                new_seg = seg if new_dir == d_i_r else seg + 1
                if (nx, ny) == end:
                    ans.append(new_seg)
                    continue
                if grid[nx][ny] != 'X':
                    in_queue.add((nx, ny, i))
                    queue.append((nx, ny, new_dir, new_seg))
    if len(ans) == 0:
        return -1
    else:
        return min(ans)
board_num = 1
```

基本信息

#:	47534007
题目:	02802
提交人:	佟永鑫
内存:	4672kB
时间:	70ms
语言:	Python3
提交时间:	2024-12-03 15:50:58

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。

前面的dp题目都很简答，最后两题自己写一直RE，看群可能是输入处理的问题，这周事情比较多，先把答案抄在这儿后面再认真看。