

Assignment #A: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Compiled by 李振硕、院系 信息管理系

1. 题目

LuoguP1255 数楼梯

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


代码：

测试点信息

源代码

源代码

```
N=int(input())
directions=[1,2]
def all_steps3(N):
    dp=[0]*(N+1)
    dp[0]=1
    for i in range(N):
        for x in directions:
            if i+x<=N:
                dp[i+x]+=dp[i]
    return dp[N]
print(all_steps3(N))
```

 sot10130

所属题目P1255 数楼梯

评测状态Accepted

提交时间2024-11-29 17:00:41

27528: 跳台阶

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

代码：

#47461456提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态：Accepted

源代码

```
n=int(input())
def all1_steps(n):
    dp=[0]*(n+1)
    dp[0]=1

    for i in range(1,n+1):
        for j in range(1,i+1):
            dp[i]+=dp[i-j]

    return dp[n]

print(all1_steps(n))
```

基本信息

#：47461456
题目：27528
提交人：24n2300093007
内存：3624kB
时间：33ms
语言：Python3
提交时间：2024-11-29 15:22:40

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

474D. Flowers

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

代码：

By sot10130, contest: Codeforces Round 271 (Div. 2), problem: (D) Flowers, [Accepted](#), #, [Copy](#)

```
MOD = 1000000007

def solve_flowers(t, k, queries):
    max_b = max(b for _, b in queries)
    dp = [0] * (max_b + 1)
    dp[0] = 1

    for i in range(1, max_b + 1):
        dp[i] = dp[i - 1]
        if i >= k:
            dp[i] = (dp[i] + dp[i - k]) % MOD

    prefix = [0] * (max_b + 1)
    for i in range(max_b + 1):
        prefix[i] = (prefix[i - 1] + dp[i]) % MOD if i > 0 else dp[i]

    results = []
    for a, b in queries:
        result = (prefix[b] - (prefix[a - 1] if a > 0 else 0) + MOD) % MOD
        results.append(result)

    return results

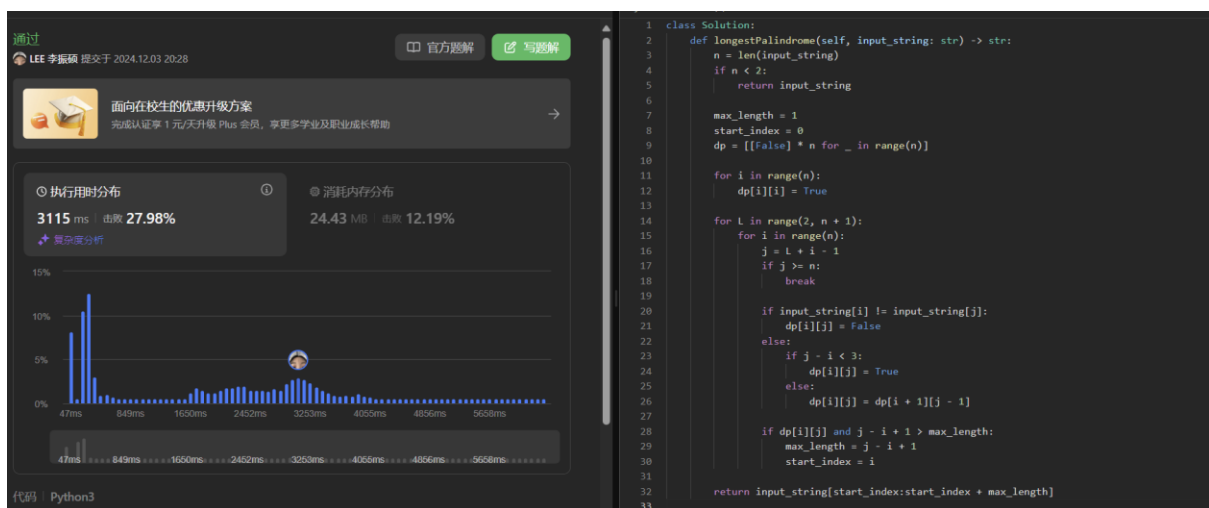
if __name__ == "__main__":
    t, k = map(int, input().split())
    queries = []
    for _ in range(t):
        a, b = map(int, input().split())
        queries.append((a, b))
    results = solve_flowers(t, k, queries)
    for res in results:
        print(res)
```

LeetCode5.最长回文子串

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

思路：

代码：



12029: 水淹七军

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

思路：真不会。。。很难受

描述

随着最后通牒的递出，C国的总攻也开始了，由于C国在地形上的优势，C国总司令下令采用水攻，剿灭A国最后的有生力量。

地形图是一个M*N的矩阵，矩阵上每一个点都对应着当前点的高度。C国总司令将选择若干个点进行放水。根据水往低处流的特性，水可以往四个方向的流动，被淹的地方的水面高度便和放水点的高度一样。然而，A国不是一马平川的，所以总会有地方是淹不到的。你的任务很简单，判断一下A国司令部会不会被淹/没掉。

我们将给你完整的地形图，然后给出A国司令部所在位置，给出C国将在哪几个点进行放水操作。你所需要的，就是给出A国司令部会不会被水淹。

输入

第一行：一个整数K，代表数据组数。

对于每一组数据：

第1行：符合题目描述的两个整数，M(0 < M <= 200)、N(0 < N <= 200)。

第2行至M+1行：每行N个数，以空格分开，代表这个矩阵上的各点的高度值H(0 <= H <= 1000)。

第M+2行：两个整数I(0 < I <= M)、J(0 < J <= N)，代表司令部所在位置。

第M+3行：一个整数P(0 < P <= M * N)，代表放水点个数。

第M+4行至M+P+4行：每行两个整数X(0 < X <= M)、Y(0 < Y <= N)，代表放水点。

输出

对于每组数据，输出一行，如果被淹则输出Yes，没有则输出No。

样例输入

```
1
5 5
1 1 1 1 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
1 1 1 1 1
3 3
2
1 1
2 2
```

样例输出

添加于 2024-11-25

提交次数 1658

尝试人数 157

通过人数 139

你的提交记录

#	结果	时间
22	Runtime Error	2024-12-03
21	Runtime Error	2024-12-03
20	Runtime Error	2024-12-03
19	Runtime Error	2024-12-03
18	Runtime Error	2024-12-03
17	Runtime Error	2024-12-03
16	Runtime Error	2024-12-03
15	Runtime Error	2024-12-03
14	Runtime Error	2024-12-03
13	Runtime Error	2024-12-03
12	Runtime Error	2024-12-03
11	Runtime Error	2024-12-03
10	Compile Error	2024-12-03
9	Runtime Error	2024-12-03
8	Runtime Error	2024-12-03
7	Runtime Error	2024-11-29
6	Runtime Error	2024-11-29
5	Runtime Error	2024-11-29
4	Runtime Error	2024-11-29
3	Runtime Error	2024-11-29
2	Runtime Error	2024-11-29
1	Compile Error	2024-11-29

状态: Accepted

源代码

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

def is_within_bounds(x, y, m, n):
    return 0 <= x < m and 0 <= y < n

def simulate_water_flow(start_x, start_y, start_height, m, n, terrain,
    directions_x = [-1, 1, 0, 0]
    directions_y = [0, 0, -1, 1]
    queue = deque([(start_x, start_y, start_height)])
    water_flow[start_x][start_y] = start_height

    while queue:
        x, y, height = queue.popleft()
        for i in range(4):
            nx, ny = x + directions_x[i], y + directions_y[i]
            if is_within_bounds(nx, ny, m, n) and terrain[nx][ny] < height:
                if water_flow[nx][ny] < height:
                    water_flow[nx][ny] = height
                    queue.append((nx, ny, height))

def main():
    data = input().split()
    idx = 0
    test_cases = int(data[idx])
    idx += 1
    results = []

    for _ in range(test_cases):
        m, n = map(int, data[idx:idx + 2])
        idx += 2
        terrain = []
        for i in range(m):
            terrain.append(list(map(int, data[idx:idx + n])))
            idx += n
        water_flow = [[0] * n for _ in range(m)]

        target_x, target_y = map(int, data[idx:idx + 2])
        idx += 2
        target_x, target_y = target_x - 1, target_y - 1

        num_water_points = int(data[idx])
        idx += 1

        for _ in range(num_water_points):
            source_x, source_y = map(int, data[idx:idx + 2])
            idx += 2
            source_x, source_y = source_x - 1, source_y - 1
            if terrain[source_x][source_y] <= terrain[target_x][target_y]:
                continue
            simulate_water_flow(source_x, source_y, terrain[source_x][source_y], m, n, terrain, water_flow)

            results.append("Yes" if water_flow[target_x][target_y] > 0 else "No")

    sys.stdout.write("\n".join(results) + "\n")

if __name__ == "__main__":
    main()
```

基本信息

#: 47541468
题目: 12029
提交人: 24n2300093007
内存: 6140kB
时间: 270ms
语言: Python3
提交时间: 2024-12-03 21:21:23

(这是别人的答案)

02802: 小游戏

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

思路：这道题也不会

代码：

状态: Accepted

源代码

```
import heapq
num1 = 1
while True:
    w, h = map(int, input().split())
    if w == 0 and h == 0:
        break
    print(f"Board #{num1}:")
    matrix = [[" "]*(w+2)] + [list(input()) + [" "] for _ in range(h)]
    dir = [(0, 1), (0, -1), (1, 0), (-1, 0)]
    num2 = 1
    while True:
        x1, y1, x2, y2 = map(int, input().split())
        if x1 == 0 and x2 == 0 and y1 == 0 and y2 == 0:
            break
        queue, flag = [], False
        in_queue = set()
        heapq.heappush(queue, (0, x1, y1, -1))
        matrix[y2][x2] = " "
        in_queue.add((-1, x1, y1))
        while queue:
            step, x, y, dirs = heapq.heappop(queue)
            if x == x2 and y == y2:
                flag = True
                break
            for i, (dx, dy) in enumerate(dir):
                px, py = x + dx, y + dy
                if 0 <= px <= w + 1 and 0 <= py <= h + 1 and (i, px, py) not in in_queue:
                    in_queue.add((i, px, py))
                    heapq.heappush(queue, (step + (dirs != i), px, py, i))
        if flag:
            print(f"Pair {num2}: {step} segments.")
        else:
            print(f"Pair {num2}: impossible.")
        matrix[y2][x2] = "X"
        num2 += 1
    print()
    num1 += 1
```

基本信息

#: 47541377
题目: 02802
提交人: 24n2300093007
内存: 4656kB
时间: 79ms
语言: Python3
提交时间: 2024-12-03 21:17:36

代码运行截图 <mark> (至少包含有"Accepted") </mark>

(这是别人的答案)

2. 学习总结和收获

这次作业除了前两个题，对我来说非常难。。。感觉需要做很多别的题。。。