

Assignment #A: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Compiled 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

LuoguP1255 数楼梯

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

思路：一个比较简单的 dp 题，但要注意 dp 必须是【0】* (n+2)，不然 $dp[2]=2$ 就没法成立（20min）

代码：

```
n=int(input())
dp=[0]*(n+2)
def climb_stairs(n):
    dp[0]=0
    dp[1]=1
    dp[2]=2
    for i in range(3,n+1):
        dp[i]=dp[i-1]+dp[i-2]
    return dp[n]
print(climb_stairs(n))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

编程语言	代码长度	用时	内存
Python 3	190B	160ms	4.90MB

 EuphoriaJ

所属题目

P1255 数楼梯

评测状态

Accepted

提交时间

2024-11-29 11:56:43

27528: 跳台阶

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

思路：与第一题类似，只是递归情况从 `dp[i-1]` 延伸到了从第 0 级台阶开始跳而已，因此需要再添加一个 for 循环（10min）

代码：

```
n=int(input())
dp=[1]*(n+2)
def climb_complex_stairs(n):
    dp[0]=0
    dp[1]=1
    dp[2]=2
    for k in range(3,n+1):
        for i in range(1,k):
            dp[k]+=dp[k-i]
    return dp[n]
print(climb_complex_stairs(n))
```

代码运行截图 ==（至少包含有"Accepted"）==

#47459060提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
n=int(input())
dp=[1]*(n+2)
def climb_complex_stairs(n):
    dp[0]=0
    dp[1]=1
    dp[2]=2
    for k in range(3,n+1):
        for i in range(1,k):
            dp[k]+=dp[k-i]
    return dp[n]
print(climb_complex_stairs(n))
```

©2002-2022 POJ 京ICP备20010980号-1

基本信息

#: 47459060
题目: 27528
提交人: EuphoriaJ
内存: 3648kB
时间: 29ms
语言: Python3
提交时间: 2024-11-29 12:05:00

[English](#) [帮助](#) [关于](#)

474D. Flowers

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

思路：做这道题开始没有取模，数据就会溢出。大致思路就是用 dp 数组定义有 i 朵花时的吃法，然后对于小于 k 朵花，只能吃红花，1 种方法

若大于 k 朵花，则可分为 dp[i-1] 与 dp[i-k] 两种情况相加，从 a 到 b 只用将 pre_sum 里的对应位置的两个数相减即可。

一定要注意取模（开始算 dp 与 pre_sum 就取了模，导致后面可能出现负数，最后加上模再除 MO 才对（2h）

代码:

```
MOD=10**9+7

MAXN=100001

dp=[0]*MAXN

pre_sum=[0]*MAXN

def count(k):

    dp[0]=1

    for i in range(1,k):

        dp[i]=1

        pre_sum[i]=(pre_sum[i-1]+dp[i])%MOD

    for i in range(k,MAXN):

        dp[i]=(dp[i-1]+dp[i-k])%MOD

        pre_sum[i]=(pre_sum[i-1]+dp[i])%MOD

def sum_up(m):

    return pre_sum[m]

t,k=map(int,input().split())

count(k)

for _ in range(t):

    a,b=map(int,input().split())

    print((sum_up(b)-sum_up(a-1)+MOD)%MOD)
```

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

Who	Problem	Lang	Verdict	Time	Me
phoriaJ	474D - Flowers	Python 3	Accepted	624 ms	740

LeetCode5.最长回文子串

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

思路：用 dp 来解决问题， $dp[i][j]$ 表示从起点 i 到终点 j 是否是回文字符串，先假定 $dp[i][i]=1$ ，即长度为 1 的都是回文字符串，然后通过起点与长度讨论其它情况，对长度为 2 加以讨论，然后再说明最一般的情况，并通过 results 列表不断更新。（1h）

代码:

```
class Solution:
```

```
def longestPalindrome(self, s: str) -> str:
```

```
n=len(s)
```

```
dp=[[0]*n for _ in range(n)]
```

```
for i in range(n):
```

```
dp[i][i]=1
```

```
result=s[i]
```

```
for j in range(n-1):
```

```
if s[j]==s[j+1]:
```

```
dp[j][j+1]=1
```

```
result=s[j:j+2]
```

```
for l in range(3,n+1):
```

```
for r in range(n-l+1):
```

$$q=r+1-1$$

```
if s[r]==s[q] and dp[r+1][q-1]:
```

```
dp[r][q]=1
```

```
result=s[r:q+1]
```

```
return result
```


代码运行截图 <mark>（至少包含有"Accepted"）</mark>

题目描述 | 通过 × | 题解 | 提交记录

全部提交记录

通过

Festive JohnsonGPY 提交于 2024.11.30 10:20

官方题解

写题解

限时优惠

PLUS 量更大

「感恩季」限时福利！
买 1 得 2 同时可享力扣 365 天 Plus 会员 + 热门平台会员权益。

→

⌚ 执行用时分布

2297 ms | 击败 47.50%

📊 复杂度分析

💾 消耗内存分布

24.54 MB | 击败 11.17%

15%

10%

12029: 水淹七军

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

思路：主要分成几个板块：

1.is_valid 函数判断 (x,y) 是否在区域内

2.bfs 函数，与之前的 bfs 模块有相同的地方，同样是通过一个布尔值的矩阵来标记，并取出一个 x,y ，再遍历周边四个 nx,ny ，如果符合便添加至 q

3.main 函数，数据输入部分，借鉴了其他同学的方法，将数据直接一次性以 **data** 的形式输入，最后再从 **data** 里面读取即可（一定要注意原区域与实际矩阵的索引值要相差 1！！！！）

最后把 **results** 里的结果分行输出

4.调用主函数

5.还要注意缩进！！（耗时至少半天）

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, visited):

    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
    visited[start_x][start_y] = True
```

```
    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 not
visited[nx][ny] and h[nx][ny] < height:

                water_height[nx][ny] = height

                visited[nx][ny] = True

                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)]
```

```
    visited = [[False] * 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] and not visited[x][y]:
            bfs(x, y, h[x][y], m, n, h, water_height,
visited)
```

```
        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 <mark>（至少包含有"Accepted"）</mark>

题目

排名

状态

提问

#47477727提交状态

查看

提交

统计

提问

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, visited):
    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
    visited[start_x][start_y] = True

    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 not visited[nx][ny] and h[nx][ny] > height:
                water_height[nx][ny] = height
                visited[nx][ny] = True
                q.append((nx, ny, height))
```

基本信息

#: 47477727

题目: 12029

提交人: EuphoriaJ

内存: 6356kB

时间: 67ms

语言: Python3

提交时间: 2024-11-30 13:08:15

02802: 小游戏

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

思路: 主要分为 bfs 函数部分与主程序 main 部分

bfs 函数: 与一般 bfs 类似 (两个核心: 一个是 visited 函数的标注, 第二个是 quene 的添加), 只是多了对于方向的讨论 (用 enumerate 同时确定位置和方向)

main 部分：输入数据+外面增加保护圈+确定 start 与 end 的位置并进行 bfs 调用(新方法：用 broad_num 来表示矩形板))(半天)

代码：

```
from collections import deque
```

```
def bfs(start, end, grid, h, w):  
    queue = deque([start])  
    visited = 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 visited):

            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':

                visited.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 <mark>（至少包含有"Accepted"）</mark>

#47478982提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

源代码

```
from collections import deque

def bfs(start, end, grid, h, w):
    queue = deque([start])
    visited = 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
            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':
                visited.add((nx, ny, i))
```

基本信息

#: 47478982
题目: 02802
提交人: EuphoriaJ
内存: 4888kB
时间: 71ms
语言: Python3
提交时间: 2024-11-30 14:12:0

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>

前两个题迅速 AC 让我怀疑自己水平有所进步，但是，第三个题一直
超时，心态有点小崩，第四个题开始没想到能这么做（虽然看后能很
快写出来），56 两个题，呵呵，

反正是做红温了，要是在考试中遇到这种题直接跳过（当然也有可能
根本做不到这里来 qwq），倒是从最后两个题中学到了一些输入与输
出的方法

感觉自己完全不靠 AI 写出来跟看懂代码完全是两个世界（一些错误
如果没有 AI 去发现真的很难处理）这种情况有什么好的解决办法
吗？

最近又回头去把贪心，矩阵，dp,dfs 与 bfs 的 pdf 上的经典例题看
了一遍，发现自己除了会套模版还是什么都不会。。。。

感觉期末是要寄了啊啊啊啊啊啊啊啊啊啊啊啊