

# Assignment #A: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Compiled by <mark>祁黄奕 城市与环境学院</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: 建立  $n+1$  长度的 dp 列表, 当  $n=0$  或  $n=1$  时只有一种上台阶方法。  $n \geq 2$  时, 当前阶的方法数等于前一阶的方法数加上前两阶的方法数之和。

代码:

```
```python
```

```
def step(n):  
    if n==0:  
        return 1  
    elif 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]
m=int(input())
print(step(m))
```

```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

洛谷 / 评测记录 / 评测详情

R192414059 记录详情

编程语言
Python 3

代码长度
231B

用时
157ms

内存
33MB

测试点信息
源代码

源代码
复制

```

def step(n):
    if n==0:
        return 1
    elif 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]
m=int(input())
print(step(m))

```

qhy0728

所属题目
P1255 数楼梯

评测状态
Accepted

提交时间
2024-12-02 10:43:22

### 27528: 跳台阶

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

思路：dp 数组，到达第 n 台阶的方法数等于到达 n-1 级台阶的方法数加上到达 n-2 级台阶的方法数，依此类推，直到到达地面（即第 0 级台阶）。

代码：

```

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

```

```

代码运行截图 ==（至少包含有"Accepted"）==

#47519173提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

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

基本信息

#: 47519173  
题目: 27528  
提交人: qhy  
内存: 3604kB  
时间: 35ms  
语言: Python3  
提交时间: 2024-12-02 17:55:04

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

### 474D. Flowers

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

思路:

代码:

```python

```
MAX = 1000000007
t, k = map(int, input().split())
MOD = int(1e9+7)
MAXN = 100001
dp = [0]*MAXN
s = [0]*MAXN
dp[0] = 1
s[0] = 1
for i in range(1, MAXN):
    if i >= k:
        dp[i] = (dp[i-1]+dp[i-k]) % MOD
    else:
        dp[i] = dp[i-1] % MOD
    s[i] = (s[i-1]+dp[i]) % MOD

for _ in range(t):
    a, b = map(int, input().split())
    print((s[b]-s[a-1]+MOD) % MOD)
```

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

<a href="#">294359171</a>	Dec/02/2024 18:49 UTC+8	qhy06	<a href="#">D - Flowers</a>	Python 3	Accepted	640 ms	7400 KB
---------------------------	----------------------------	-------	-----------------------------	-------------	----------	--------	---------

### ### LeetCode5. 最长回文子串

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

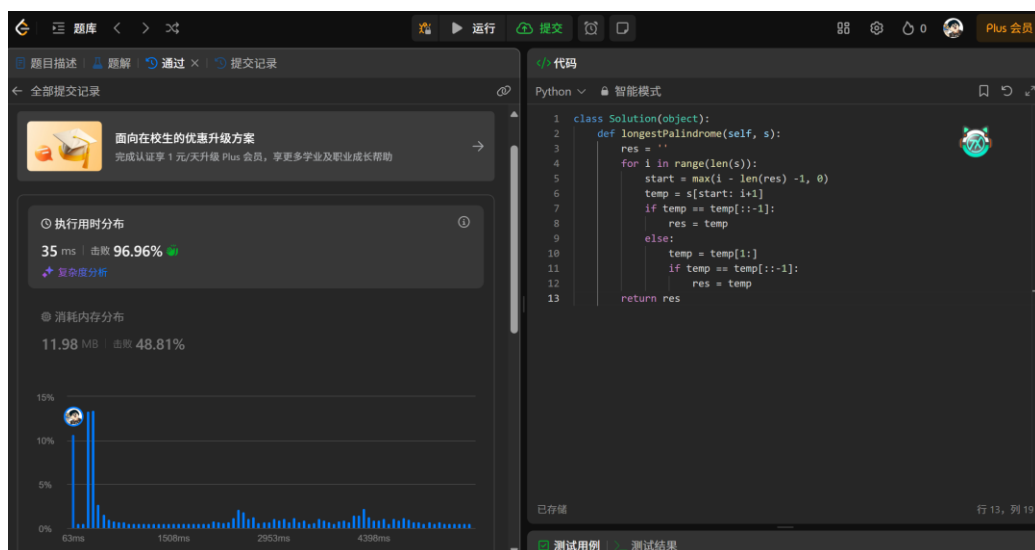
思路：在处理回文子串问题时，需要考虑奇数和偶数两种情况。首先设定一个空字符串作为最大回文子串 `res = ''`，然后逐步更新。遍历数组，使用指针 `i` 向前寻找长度大于当前 `res` 的回文子串，注意定位到开头的前一位 `start = max(0, i-len(s)-1)`。根据索引范围切片判断是否为回文子串，分为奇数和偶数两种情况。命中条件则更新 `res`，最终得到最长回文子串。

代码：

```
```python
```

```
class Solution(object):
    def longestPalindrome(self, s):
        res = ''
        for i in range(len(s)):
            start = max(i - len(res) - 1, 0)
            temp = s[start: i+1]
            if temp == temp[::-1]:
                res = temp
            else:
                temp = temp[1:]
                if temp == temp[::-1]:
                    res = temp
        return res
```
```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>



### 12029: 水淹七军

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

思路:

代码:

```python

```
from collections import deque
import sys
MAXN=1111
def bfs(x,y):
    global slb,mat
    q=deque([(x,y)])
    d=[(-1,0),(0,1),(1,0),(0,-1)]
    while q:
        x,y=q.popleft()
        for dx,dy in d:
            nx,ny=x+dx,y+dy
            if 1<=nx<=m and 1<=ny<=n \
            and mat[nx][ny]<mat[x][y]:
                q.append((nx,ny))
                mat[nx][ny]=mat[x][y]
                if (nx,ny)==slb:
                    return('Yes')
    return ('No')
lines=list(sys.stdin.read().split())

t=int(lines[0])
id=1
for _ in range(t):
    m,n=map(int,(lines[id],lines[id+1]))
    mat=[ [MAXN]*(n+2) ]
    id+=2
    for i in range(m):
        mat.append([MAXN]+list(map(int,lines[id:id+n]))+[MAXN])
        id+=n
    mat.append([MAXN]*(n+2))
    slb=(int(lines[id]),int(lines[id+1]))
    id+=2
    k=int(lines[id])
    id+=1
    H2O=[]
```

```
for i in range(k):
    H2O.append((int(lines[id]),int(lines[id+1])))
    id+=2
for h20 in H2O:
    ans=bfs(h20[0],h20[1])
    if ans=='Yes':
        print(ans)
        break
    else:print('No')
'''
```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

#47521489提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque
import sys
MAXN=1111
def bfs(x,y):
    global slb,mat
    q=deque([(x,y)])
    d=[(-1,0),(0,1),(1,0),(0,-1)]
    while q:
        x,y=q.popleft()
        for dx,dy in d:
```

基本信息

#: 47521489

题目: 12029

提交人: qhy

内存: 6148kB

时间: 54ms

语言: Python3

提交时间: 2024-12-02 20:10:30

### 02802: 小游戏

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

思路：可以用递归方法求解通过穷举方式找到从起点到终点的路径，朝一个方向走下去，如果走不通，则换一个方向走；四个方向都走不通，则回到上一步的地方，再换一个方向走；依次走下去，直到走到终点。该问题的路径只包含水平或者竖直的直线段，所以需要记录每一步走的方向。如果上一步走的方向和这一步走的方向相同，递归搜索时路径数不变，否则路径数加 1。

另外根据题意，路径只包含水平或者竖直的直线段，路径不能穿过别的游戏卡片，但是允许路径临时离开矩形板，所以可以在矩形板最外层增加一圈格子，路径可以通过这些新增加的格子可以设置一个二维数组来表示迷宫(空格表示无游戏卡片，X 表示有游戏卡片)，用另一个二维数组来标记某个位置的格子是否已经被走过了。

代码：

```
'''python
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()
    # print(x,y,end)
    if (x, y) == end:
        # return seg
        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:
                # return new_seg
                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)] + \
    # [[' ']+list(input())+[' '] for _ in range(h)] + \
    # [[' ']*(w+2)]
    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>

#47522159提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

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()
        # print(x,y,end)
        if (x, y) == end:
            # return seg
            ans.append(seg)
            break

```

基本信息

#: 47522159  
 题目: 02802  
 提交人: qhy  
 内存: 4644kB  
 时间: 69ms  
 语言: Python3  
 提交时间: 2024-12-02 20:39:16

## ## 2. 学习总结和收获

这次做下来依然要感叹好难好难好难！打算好好啃一下 dp 的 easy 和 middle 题目，熟悉模板，期末遇到觉得难的题目（特别是像最后两题，做作业的时候看完 ai 和题解还是不懂……）会直接放弃，显然需要再多做点题，并训练一下数学思维。日常希望能过！