

Assignment #10: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Compiled by <mark>陶嘉瑞-物理学院</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:
求出递推关系即可

代码:

```
```python
n=int(input())
dp=[0]*(n+1)

if n>=3:
 dp[1]=1
 dp[2]=2
 for i in range(3,n+1):
 dp[i]=dp[i-1]+dp[i-2]
 print(dp[n])
else:
 print(n)

```
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

talenttao

| | |
|------|---------------------|
| 所属题目 | P1255 数楼梯 |
| 评测状态 | Accepted |
| 提交时间 | 2024-11-26 18:40:38 |

27528: 跳台阶

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

思路：
找出递推关系

代码：

```
```python
n=int(input())
dp=[1]*n
for i in range(1,n):
 for j in range(0,i):
 dp[i]+=dp[j]
print(dp[-1])
```
```

代码运行截图 ==（至少包含有"Accepted"）==
状态: Accepted

源代码

```
n=int(input())
dp=[1]*n
for i in range(1,n):
    for j in range(0,i):
        dp[i]+=dp[j]
print(dp[-1])
```

基本信息

#: 47412284
题目: 27528
提交人: talenttao
内存: 3616kB
时间: 34ms
语言: Python3
提交时间: 2024-11-26 18:45:39

474D. Flowers

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

思路：

找出递推关系

代码：

```
```python
t,k=map(int,input().split())
mod=10**9+7
n=100001
dp=[0]*100001
dp[0]=1
s=[0]*100001
for i in range(1,100001):
 if i>=k:
 dp[i]=(dp[i-1]+dp[i-k])%mod
 else:
 dp[i]=1
 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)
```

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

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

```
t,k=map(int,input().split())
mod=10**9+7
n=100001
dp=[0]*100001
dp[0]=1
s=[0]*100001
for i in range(1,100001):
    if i>=k:
        dp[i]=(dp[i-1]+dp[i-k])%mod
    else:
        dp[i]=1
    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)
```

LeetCode5.最长回文子串

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

思路：

利用二维数组进行动态规划

代码：

```python

class Solution:

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

n = len(s)

if n < 2:

return s

num = 1

begin = 0

dp = [[False] \* n for \_ in range(n)]

for i in range(n):

dp[i][i] = True

for length in range(2, n + 1):

for i in range(n):

j = length + i - 1

if j >= n:

break

if s[i] != s[j]:

dp[i][j] = False

else:

if j - i < 3:

dp[i][j] = True

else:

dp[i][j] = dp[i + 1][j - 1]

if dp[i][j] and j - i + 1 > num:

num = j - i + 1

begin = i

return s[begin:begin + num]

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

通过

Boring PayneB8L 提交于 2024.11.26 20:42

官方题解

写题解



「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



3251 ms | 击败 22.73%

🌟 复杂度分析

💾 消耗内存分布

24.60 MB | 击败 10.75%

12029: 水淹七军

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

思路:

本题原先第一遍自己写的时候有思路也写出来了,但是由于还不会 sys 输入所以就参考了答案

代码:

```
```python
import sys

sys.setrecursionlimit(300000)
input = sys.stdin.read

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

def dfs(x, y, water_height_value, m, n, h, water_height):
 dx = [-1, 1, 0, 0]
 dy = [0, 0, -1, 1]

 for i in range(4):
 nx, ny = x + dx[i], y + dy[i]
 if is_valid(nx, ny, m, n) and h[nx][ny] < water_height_value:
 if water_height[nx][ny] < water_height_value:
```

```

 water_height[x][y] = water_height_value
 dfs(nx, ny, water_height_value, m, n, h, water_height)

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

 dfs(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")
'''

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
import sys

sys.setrecursionlimit(300000)
input = sys.stdin.read

def is valid(x, y, m, n):
```

基本信息

#: 47429904  
题目: 12029  
提交人: talenttao  
内存: 15920kB  
时间: 317ms  
语言: Python3  
提交时间: 2024-11-27 18:17:35

### ### 02802: 小游戏

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

思路:

利用 bfs 搜索

代码:

```
```python
from collections import deque
def bfs(start,end,maze):
    q=deque([start])
    inq=set()
    ans=[]
    while q:
        seg,(x0,y0),dir=q.popleft()
        if (x0,y0)==end:
            ans.append(seg)
            break
        for i in range(4):
            x=x0+dx[i]
            y=y0+dy[i]
            if 0 <=x<w+2 and 0<=y< h+2 and ((x,y,i) not in inq):
                new_dir=i
                new_seg=seg if new_dir==dir else seg+1
                if (x,y)==end:
                    ans.append(new_seg)
                    continue
                if maze[y][x]!='X':
                    inq.add((x,y,i))
                    q.append((new_seg,(x,y),new_dir))

    if len(ans)==0:
        return -1
```

```

dx = [0, 0, 1, -1]
dy = [1, -1, 0, 0]
s=1

while True:
    w,h=map(int,input().split())
    if w==h==0:
        break
    print('Board #{}:'.format(s))
    maze=[' '*(w+2)]+[ ' '+input()+ ' ' for _ in range(h)]+[ ' '*(w+2)]
    n=1
    while True:
        x1,y1,x2,y2=map(int,input().split())
        if x1==x2==y1==y2==0:
            break
        start = (0,(x1, y1), -1)
        end = (x2, y2)
        num=bfs(start,end,maze)
        if num>0:
            print('Pair {}: {} segments.'.format(n,num))
        else:
            print('Pair {}: impossible.'.format(n))
        n+=1
    print()
    s+=1

```

```
#: 47434974
题目: 02802
提交人: talenttao
内存: 4620kB
时间: 75ms
语言: Python3
提交时间: 2024-11-27 22:24:41
```


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>

感觉本周题目难度参差不齐前面的几题较为简单，而后面的两题花了我将近半天时间。其中最后一题先是看反了横纵坐标，之后又因为输出格式问题导致了 presentation error。感觉这些题目其实都有思路，但是在细节上却频频出错。因此之后的每日练习中要多注重细节。