

Assignment #10: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Compiled by 熊程宇 物理学院

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

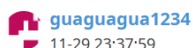
dp, bfs, <https://www.luogu.com.cn/problem/P1255>

思路:

代码:

```
n = int(input())
dp = [0 for _ in range(n + 1)]
dp[0] = 1
dp[1] = 1
for i in range(2, n + 1):
    dp[i] = dp[i - 1] + dp[i - 2]
print(dp[-1])
```

代码运行截图 (至少包含有"Accepted")



11-29 23:37:59



100

P1255 数楼梯

160ms / 4.88MB / 147B Python 3

27528: 跳台阶

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

思路:

代码:

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

基本信息

#: 47471893
题目: 27528
提交人: 24n2400011504
内存: 3624kB
时间: 37ms
语言: Python3
提交时间: 2024-11-29 23:40:26

©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:

首先查询数学知识: m个A和n个B的不同排列有几种

查询得可以用递归方法解决, 于是更新规则就出来了

交了四次是因为dp应该开 $1e5+1$ 而不是 $1e5$, 本质上是没想明白dp可以引入一个dp[0]来简化代码。

代码:

```
t, k = map(int, input().split())
dp = [0 for _ in range(10 ** 5 + 1)]
for i in range(k):
    dp[i] = 1
for i in range(k, 10 ** 5 + 1):
    dp[i] = dp[i - 1] % 1000000007 + dp[i - k] % 1000000007
    dp[i] % 1000000007
sum_dp = [0 for _ in range(10 ** 5 + 1)]
sum_dp[0] = 1
for i in range(1, 10 ** 5 + 1):
    sum_dp[i] = sum_dp[i - 1] + dp[i]
```

```
for i in range(t):
    a, b = map(int,input().split())
    print((sum_dp[b] - sum_dp[a-1]) % 1000000007)
```

代码运行截图 (至少包含有"Accepted")

→ Last submissions		
Submission	Time	Verdict
294153761	Dec/01/2024 10:12	Accepted
294153376	Dec/01/2024 10:09	Wrong answer on test 34
294153125	Dec/01/2024 10:07	Runtime error on test 5
294152990	Dec/01/2024 10:05	Runtime error on test 5

LeetCode5.最长回文子串

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

思路:

写了一串又臭又长的代码还不过后决定寻求GPT的帮助

代码:

```
class Solution:
    def longestPalindrome(self, s: str) -> str:
        def f(a, b):
            while a >= 0 and b < len(s) and s[a] == s[b]:
                a -= 1
                b += 1
            return s[a + 1:b]

        r = ""
        for i in range(len(s)):
            x = f(i, i)
            y = f(i, i + 1)
            if len(x) > len(r): r = x
            if len(y) > len(r): r = y
        return r
```

代码运行截图 (至少包含有"Accepted")



12029: 水淹七军

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

思路:

刨开读入读出的问题，淹没条件：该点坐标比放水点的坐标低，而不是比临近的坐标低！

代码:

```
import sys
from collections import deque

dx=[0,1,0,-1]
dy=[-1,0,1,0]
input = sys.stdin.read
a=input().split()
idx=0
k=int(a[idx])
idx+=1
```

```

ans=[]
for _ in range(k):
    m,n=map(int,a[idx:idx+2])
    idx+=2
    h=[]
    hw=[[0 for i in range(n)]for j in range(m)]
    for i in range(m):
        h.append(list(map(int,a[idx:idx+n])))
        idx+=n
    tx,ty=map(int,a[idx:idx+2])
    tx,ty=tx-1,ty-1
    idx+=2
    p=int(a[idx])
    idx+=1
    for i in range(p):
        x,y=map(int,a[idx:idx+2])
        idx+=2
        x,y=x-1,y-1
        q=deque()
        q.append((x,y))
        if h[x][y]<=h[tx][ty]:
            continue
        ht=h[x][y]
        hw[x][y]=h[x][y]
        while q:
            x,y=q.popleft()
            for j in range(4):
                nx,ny=x+dx[j],y+dy[j]
                if 0<=nx<m and 0<=ny<n:
                    if ht>hw[nx][ny] and ht>h[nx][ny]:
                        hw[nx][ny]=ht
                        q.append((nx,ny))
    ans.append('Yes' if hw[tx][ty]>0 else 'No')
sys.stdout.write('\n'.join(ans)+'\n')

```

代码运行截图 (至少包含有"Accepted")

02802: 小游戏

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

思路:

记住这种读入数据的流程:

while True:

index += 1

构式读入和构式格式。

调代码时遇见的问题: x和y是反的, 求的是最小线段数而不是最小长度 (这可以从代码命名看出一些历史遗留问题) 输出没有打". ", 输出没有打空行.....

不过本身就是套广度优先的模版题，似乎在学好基础后更难题目就是把各种模块（这里拼了复杂的读入输出数据和广度优先模版）套在一块，再微调一些东西就成题目了。

嗯这很竞赛。

代码：

```
from collections import deque
index = 0
dir = [(1, 0), (-1, 0), (0, 1), (0, -1)]
while True:
    index += 1
    w, h = map(int, input().split())
    if w == 0 and h == 0:
        break
    board = [[0 for _ in range(w + 2)] for _ in range(h + 2)]
    for i in range(h):
        row = input()
        for j, char in enumerate(row):
            if char == 'X':
                board[i + 1][j + 1] = 1
    test_cases = []
    while True:
        x1, y1, x2, y2 = map(int, input().split())
        if x1 == 0 and y1 == 0 and x2 == 0 and y2 == 0:
            break
        test_cases.append((x1, y1, x2, y2))
    def min_length(test_case):
        q = deque()
        q.append((test_case[1], test_case[0], 0, -1))
        visited = [[False for _ in range(w + 2)] for _ in range(h + 2)]
        min_length = 10000
        possible = False
        while q:
            nx, ny, length, dir_n = q.popleft()
            visited[nx][ny] = True
            for i, (dx, dy) in enumerate(dir):
                cx, cy = nx + dx, ny + dy
                if (cx, cy) == (test_case[3], test_case[2]):
                    possible = True
                    if dir_n == i:
                        min_length = min(min_length, length)
                    else:
                        min_length = min(min_length, length + 1)
            for i, (dx, dy) in enumerate(dir):
                cx, cy = nx + dx, ny + dy
                if 0 <= cx < h + 2 and 0 <= cy < w + 2 and not visited[cx][cy] and
board[cx][cy] == 0 :
                    visited[cx][cy] = True
                    if i == dir_n:
                        q.append((cx, cy, length, i))
                    else:
                        q.append((cx, cy, length + 1, i))
                    visited[cx][cy] = False
        return possible, min_length
    print(f'Board #{index}:')
    for i, test_case in enumerate(test_cases):
        possible, length = min_length(test_case)[0], min_length(test_case)[1]
```

```
if possible:
    print(f'Pair {i + 1}: {length} segments.')
if not possible:
    print(f'Pair {i + 1}: impossible.')
print()
```

代码运行截图 (至少包含有"Accepted")

你的提交记录

#	结果	时间
3	Accepted	2024-12-02
2	Presentation Error	2024-12-02
1	Wrong Answer	2024-12-02

2. 学习总结和收获

这次作业的最后一题应该算难的了把.....

深度搜索和广度搜索本身比较复杂，但是套模版套习惯之后也就好了。把问题分成几个简单的小问题，写好读入读出的接口，再套模版，感觉大概就是这样吧，目前的深度搜索和广度搜索题目比需要思考的dp和贪心还是简单一些，尽管用时更长。