Assignment #10: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Complied by <mark>胡新璞，工学院</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

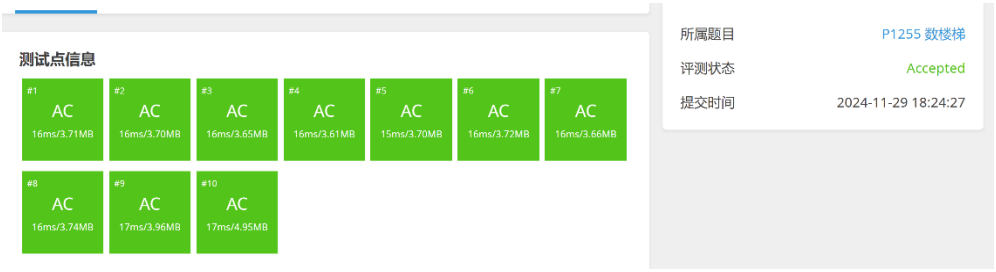
LuoguP1255 数楼梯

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

代码:

```
n = int(input())
dp = [0] * (n + 1)
if n == 1:
    print(1)
else:
    dp[0],dp[1] = 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 <mark>（至少包含有"Accepted"）</mark>



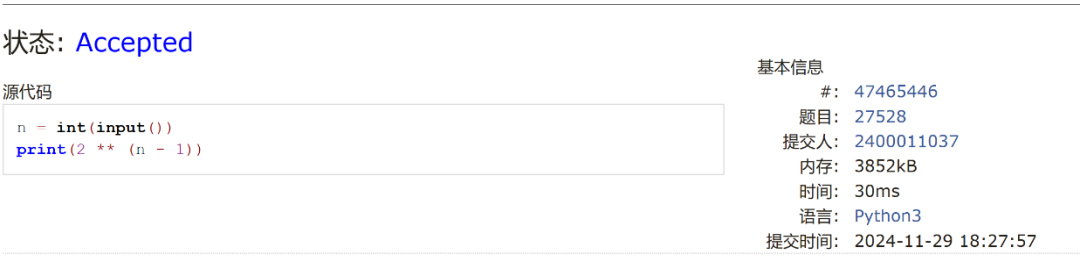
27528: 跳台阶

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

代码:（主打一个数学方法至上。不过机考的时候应该想不出数学方法罢（悲）。Dp 的话就是取前面所有项的和，应该也好写。）

```
n = int(input())
print(2 ** (n - 1))
```

代码运行截图 ==（至少包含有"Accepted"）==



474D. Flowers

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

代码：

```
t, k = map(int, input().split())
dp = [1] * 100001
mod = 10 ** 9 + 7
for i in range(k,100001):
    dp[i] = (dp[i - 1] + dp[i - k]) % mod
sum_dp = [0] * 100002
for i in range(1, 100001):
    sum_dp[i] = (sum_dp[i - 1] + dp[i - 1]) % mod
sum_dp[100001] = (sum_dp[100000] + dp[100000]) % mod
for _ in range(t):
    a,b = map(int, input().split())
    print((sum_dp[b + 1] - sum_dp[a] + mod) % mod)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>


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

```
t, k = map(int, input().split())
dp = [1] * 100001
mod = 10 ** 9 + 7
for i in range(k, 100001):
    dp[i] = (dp[i - 1] + dp[i - k]) % mod
sum_dp = [0] * 100002
for i in range(1, 100001):
    sum_dp[i] = (sum_dp[i - 1] + dp[i - 1]) % mod
sum_dp[100001] = (sum_dp[100000] + dp[100000]) % mod
for _ in range(t):
    a, b = map(int, input().split())
    print((sum_dp[b + 1] - sum_dp[a] + mod) % mod)
```

LeetCode5.最长回文子串

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

代码:

```
class Solution(object):
    def longestPalindrome(self, s):
        begin_with = 0
        max_length = 1
        for i in range(len(s)):
            l = i
            r = i + 1
            while l >= 0 and r < len(s) and s[l] == s[r]:
                l -= 1
                r += 1
            if (r - l - 1) > max_length:
                max_length = r - l - 1
                begin_with = l + 1
            l = i
            r = i
            while l >= 0 and r < len(s) and s[l] == s[r]:
                l -= 1
                r += 1
            if (r - l - 1) > max_length:
                max_length = r - l - 1
                begin_with = l + 1
        return s[begin_with:begin_with + max_length]
```

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/

代码:

```
import sys
sys.setrecursionlimit(1000000)
def dfs(matrix,matrix_,x,y):
    directions = [[0,1],[0,-1],[1,0],[-1,0]]
    for i in range(len(directions)):
        nx = x + directions[i][0]
        ny = y + directions[i][1]
        if 0 <= nx < m and 0 <= ny < n:
            if matrix[nx][ny] < matrix[x][y]:
                matrix[nx][ny] = matrix[x][y]
                matrix_[nx][ny] = True
                dfs(matrix,matrix_,nx,ny)
data = sys.stdin.read().split()
k = int(data[0])
id = 1
ans = []
for _ in range(k):
    m,n = int(data[id]),int(data[id+1])
    id += 2
    matrix = [list(map(int, data[id + (i * n):id + (i + 1) * n])) for i in range(m)]
    matrix_ = [[False for i in range(n)] for j in range(m)]
    id += m * n
    a,b = int(data[id]) - 1,int(data[id+1]) - 1
    id += 2
    p = int(data[id])
    id += 1
    for __ in range(p):
        x,y = int(data[id]) - 1,int(data[id+1]) - 1
        dfs(matrix,matrix_,x,y)
        id += 2
    ans.append("Yes" if matrix_[a][b] else "No")
sys.stdout.write("\n".join(ans) + "\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(1000000)

def dfs(matrix,matrix_,x,y):
    directions = [[0,1],[0,-1],[1,0],[-1,0]]
    for i in range(len(directions)):
        nx = x + directions[i][0]
        ny = y + directions[i][1]
```

基本信息

#: 47512084
题目: 12029
提交人: 2400011037
内存: 16028kB
时间: 329ms
语言: Python3
提交时间: 2024-12-02 11:27:04

02802: 小游戏

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

代码：

```
from collections import deque

def bfs(m,n,a,b):
    ans = []
    directions = [[0,1], [1,0], [-1,0], [0,-1]]
    q = deque([(-1,0,(m,n))])
    in_queue = {(-1,(m,n))}

    while q:
        d,s,(x,y) = q.popleft()
        if x == a and y == b:
            ans.append(s)
            break
        for i in range(len(directions)):
            nx = x + directions[i][0]
            ny = y + directions[i][1]
            if 0 <= nx < h + 2 and 0 <= ny < w + 2 and ((i, nx, ny) not in in_queue):
                nd = i
                if nd != d:
                    ns = s + 1
                else:
                    ns = s
                if nx == a and ny == b:
                    ans.append(ns)
                if matrix[nx][ny] != "X":
                    in_queue.add((nd, nx, ny))
                    q.append((nd, ns, (nx, ny)))

    if len(ans):
        return str(min(ans)) + " segments."
    else:
        return "impossible."

case_num = 0
while True:
```

```
    w,h = map(int,input().split())
    if w == 0 and h == 0:
        break
    case_num += 1
    print("Board #" + str(case_num) + ":")
    matrix = [[" "] * (w + 2) for i in range(h + 2)]
    for _ in range(1, h + 1):
        matrix[_][1:-1] = map(str, input())
    pair_num = 0

    while True:
        y1,x1,y2,x2 = map(int,input().split())
        if y1 == 0 and y2 == 0 and x1 == 0 and x2 == 0:
            break
        pair_num += 1
        situation = bfs(x1,y1,x2,y2)
        print("Pair " + str(pair_num) + ": " + situation)

    print()
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
from collections import deque

def bfs(m,n,a,b):
    ans = []
    directions = [[0,1], [1,0], [-1,0], [0,-1]]
    q = deque([(-1,0,(m,n))])
    in_queue = {(-1,(m,n))}
```

基本信息

#:	47538399
题目:	02802
提交人:	2400011037
内存:	4740kB
时间:	78ms
语言:	Python3
提交时间:	2024-12-03 19:02:36

（刚花了一个下午写明白弄明白群里的同学就给了反例，后续再修改吧）

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>

我勒个水淹七军难道真的不是把我淹死了吗？（x）
这周作业前两个题秒了，第四题写了半个多钟，各种小错，幸好 Leetcode 能看到错解方便 debug，不然得疯掉。我一开始的思路跟马拉车算法的思路是相似的，但是太菜了代码写不对，于是改成双指针。第三题因为一开始题目没看懂（后来发现少看了个 white，我这该死的英语阅读理解能力，四级感觉都要挂。）后来做的时候处理前缀和的边界没有弄好又费了一点力气。水淹七军感觉思路其实算不上难，比较常规的 dfs？但是读入实在是逆天，以及因为没有手动修改深度导致的 RE 让我对着代码浪费了好几个小时（x）。

总结是依然要提高对于边界的处理、对题目条件的转化（找方程）能力，最近发现 AI 有点难用，他给我改的双指针连 OJ 给的样例都过不去，还把我的代码说成对的把题解说成错的，即便我怎么苦口婆心的劝说它就是信誓旦旦地说自己没毛病，最后还是请教同学+自己分析才搞明白……感觉还是要自己花更大力气研究讲义和题目。