Assignment #10: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Complied by 同学的姓名、院系

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

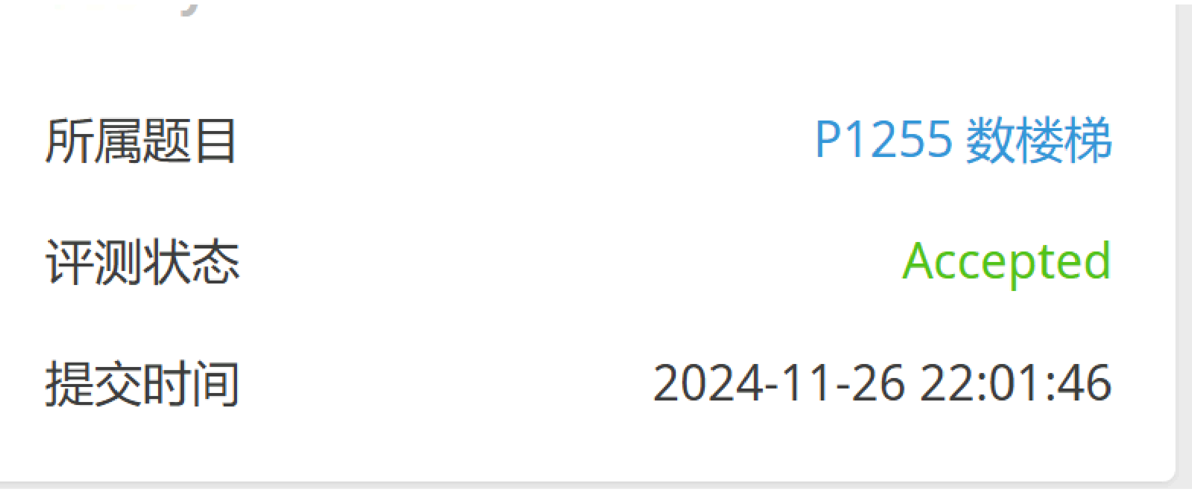
思路：

对于登上第n阶台阶的人来说，他可以踏2级前往（n + 2)， 也可以踏一级前往(n + 1)

代码：

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

代码运行截图 （至少包含有"Accepted"）



所属题目 P1255 数楼梯

评测状态 Accepted

提交时间 2024-11-26 22:01:46

27528: 跳台阶

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

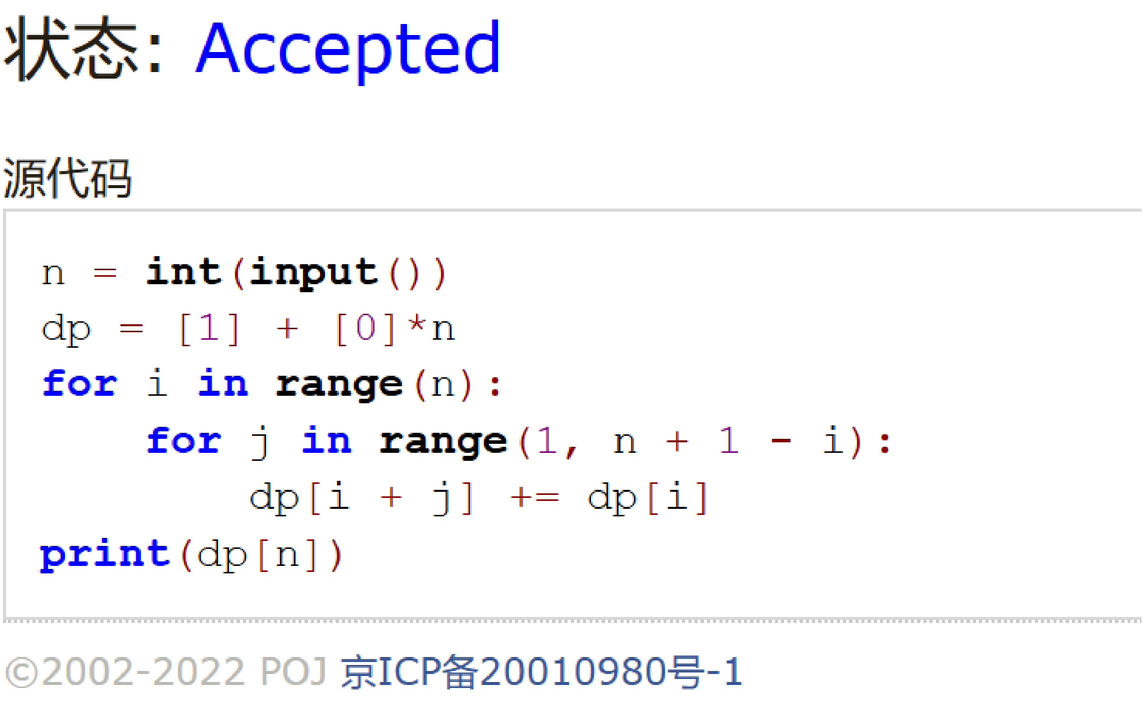
思路：

类似背包，向后面的台阶上面加方法数

代码：

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

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

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

©2002-2022 POJ 京ICP备20010980号-1

474D. Flowers

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

思路:

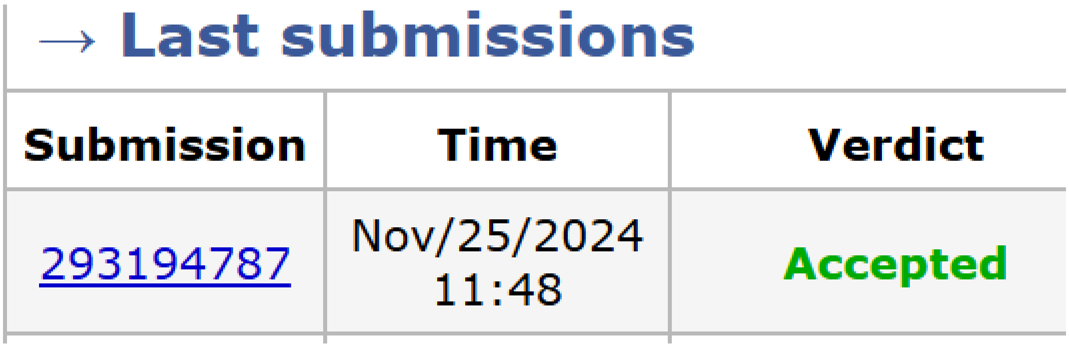
同上。

其实一开始先做的是最难版本来着......dp基本公式不难，解决前缀和用了一段时间

代码:

```
t, k = map(int, input().split())
dp = [1]*k + [0]*(10**5 + 1 - k)
s = [i for i in range(k)] + [0]*(10**5 + 1 - k)
for i in range(k, 10**5 + 1):
    dp[i] = (dp[i - 1] + dp[i - k])%(10**9 + 7)
    s[i] = (s[i - 1] + dp[i])%(10**9 + 7)
for _ in range(t):
    begin, end = map(int, input().split())
    print((s[end] - s[begin - 1] + 10**9 + 7)%(10**9 + 7))
```

代码运行截图 （至少包含有"Accepted"）



→ Last submissions

Submission	Time	Verdict
293194787	Nov/25/2024 11:48	Accepted

LeetCode5.最长回文子串

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

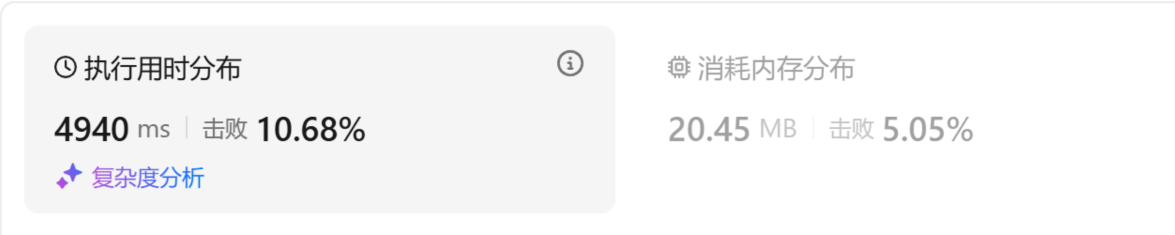
思路：

代码：

```
# O($n^2$) 4950ms
class Solution(object):
    def longestPalindrome(self, s):
        """
        :type s: str
        :rtype: str
        """
        n = len(s)
        if n == 1:
            return s
        dp = [[False for _ in range(n)] for _ in range(n)]
        max_len = 1
        start = 0
        for j in range(1, n):
            for i in range(j):
                if (j-i <= 2 and s[i] == s[j]) or (s[i] == s[j] and dp[i + 1][j - 1]):
                        dp[i][j] = True
                        cur_len = j - i + 1
                if dp[i][j]:
                    if cur_len > max_len:
                        max_len = cur_len
                        start = i
        return s[start:start + max_len]

 #加入双指针 中心扩散 335ms
def palidrome(s, l, r):
            while l >= 0 and r < len(s) and s[l] == s[r]:
                l -=1
                r +=1
            return s[l+1:r]
        max_str = ""
        max_num = 0
        for i in range(len(s)):
            dan_str = palidrome(s, i, i)
            if len(dan_str) > max_num:
                max_num = len(dan_str)
                max_str = dan_str
            ou_str = palidrome(s, i, i+1)
            if len(ou_str) > max_num:
                max_num = len(ou_str)
                max_str = ou_str
        return max_str
```

代码运行截图 （至少包含有"Accepted"）



执行用时分布

4940 ms | 击败 10.68%

复杂度分析

消耗内存分布

20.45 MB | 击败 5.05%

12029: 水淹七军

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

思路：

实在找不到错在哪里了……能运行但是RE，请老师帮忙看一看

代码：

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

D = [[1, 0], [0, 1], [-1, 0], [0, -1]]
def dfs(x, y, m, n, water):
    for ele in D:
        a_x, a_y = ele[0] + x, ele[1] + y
        if is_valid(a_x, a_y, m, n) and water > maps[a_x][a_y]:
            if h[a_x][a_y] < water:
                h[x][y] = water
                dfs(a_x, a_y, m, n, water)

k = int(input())
results = []

for _ in range(k):
    m, n = map(int, input().split())
    maps = []
    for i in range(m):
        maps.append(list(map(int, input().split())))
    h = [[0]*n for o in range(m)]

    i, j = map(int, input().split())
    i, j = i - 1, j - 1
    p = int(input())
    for u in range(p):
        x, y = map(int, input().split())
        x, y = x - 1, y - 1
        if maps[x][y] <= maps[i][j]:
            continue
        dfs(x, y, m, n, maps[x][y])

    results.append("Yes" if h[i][j] > 0  else "No")
print("\n".join(results))
```

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：

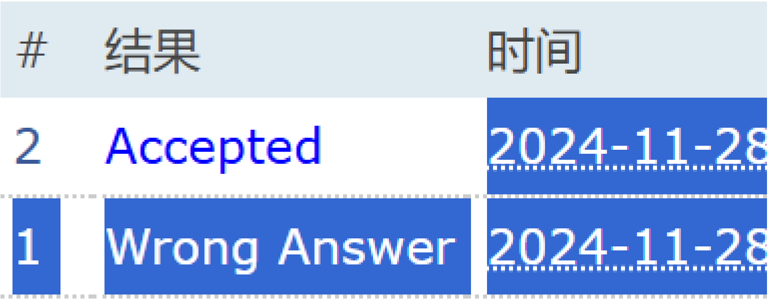
读取位置，记录一个位置是否被访问过，如果没有就继续前进，否则返回上一节点

代码：

```
import heapq

board = 1
while True:
    w, h = map(int, input().split())
    if w == 0 and h == 0:
        break
    print('Board #%d:' %(board))
    martix = [[' ']*(w + 2)]+[[' '] + list(input())+[' '] for _ in range(h)]  + [[' ']*(w + 2)]
    D = [(0,1), (0,-1), (1,0), (-1,0)]
    pair = 1
    while True:
        x1, y1, x2, y2 = map(int, input().split())
        if x1 == 0 and x2 == 0 and y1 == 0 and y2 == 0:
            break
        queue, flag = [], False
        vis = set()
        heapq.heappush(queue, (0, x1, y1, -1))
        martix[y2][x2] = ' '
        vis.add((-1, x1, y1))
        while queue:
            step, x, y, dirs = heapq.heappop(queue)
            if x == x2 and y == y2:
                flag = True
                break
            for i, (dx,dy) in enumerate(D):
                a_x, a_y = x + dx, y + dy
                if 0 <= a_x <= w + 1 and 0 <= a_y <= h + 1 and (i, a_x, a_y) not in vis and martix[a_y][a_x] != "X":
                    vis.add((i, a_x, a_y))
                    heapq.heappush(queue, (step + (dirs != i), a_x, a_y, i))
        if flag:
            print("Pair %d: %d segments." %(pair, step))
        else:
            print(f"Pair {pair}: impossible.")
        martix[y2][x2] = 'X'
        pair += 1
    print()
    board += 1
```

代码运行截图 （至少包含有"Accepted"）



#	结果	时间
2	Accepted	2024-11-28
1	Wrong Answer	2024-11-28

2. 学习总结和收获

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。

前面的dp还挺简单的，通过小游戏对bfs了解更深入了些

水淹七军至今没发现错在哪里......心累