

Assignment #10: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Compiled by <mark>汤伟杰，信息管理系</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：

斐波那契数列，使用空间复杂度低的方法来写。如果需要取模的话要在(a+b)后就取模。

代码：

```
def step(n):
    if n==1 or n==2:
        return n
    a,b=1,2
    for _ in range(n-2):
        a,b=b,a+b
    return b

print(step(int(input())))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

测试点信息

源代码

twj_ink

所属题目P1255 数楼梯

评测状态Accepted

提交时间2024-11-26 15:10:42

测试点信息

#1 AC 16ms/3.72MB	#2 AC 16ms/3.70MB	#3 AC 15ms/3.72MB	#4 AC 16ms/3.72MB	#5 AC 16ms/3.64MB	#6 AC 16ms/3.72MB	#7 AC 16ms/3.71MB
#8 AC 16ms/3.68MB	#9 AC 16ms/3.71MB	#10 AC 17ms/3.73MB				

27528: 跳台阶

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

思路:

是第一题的进阶，上一题只与前两个状态有关，这个题与之前的所有状态都有关，但是要加上直接一步上完楼梯的 1 步。

代码:

```
def step(n):
    if n==1 or n==2:
        return n
    dp=[1,2]+[0]*(n-2)
    for i in range(2,n):
        dp[i]=1+sum(dp[j] for j in range(i))
    return dp[-1]

print(step(int(input())))
```

代码运行截图 == （至少包含有"Accepted"） ==

#47406937提交状态

查看提交统计提问

状态: Accepted

源代码

```
def step(n):
    if n==1 or n==2:
        return n
    dp=[1,2]+[0]*(n-2)
    for i in range(2,n):
        dp[i]=1+sum(dp[j] for j in range(i))
    return dp[-1]

print(step(int(input())))
```

基本信息

#: 47406937

题目: 27528

提交人: 24n2400016635

内存: 3572kB

时间: 33ms

语言: Python3

提交时间: 2024-11-26 15:15:43

©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:

见第二段代码的关键式子： $dp[i]=dp[i-1]+dp[i-k]$ ，借助了 gpt 帮助才理解：对第 i 个长度，可以选择在第 $i-1$ 个长度末尾加一个 R ，或者选择在第 $i-k$ 个长度末尾加一组 W 。这时候我才发现和作业前两题的关系。。写的第一段代码感觉逻辑有点不对但是答案是对的，问 gpt 也不太理解？

代码：

```
#只构建前缀和数组，dp 数组用第一题的小空间来存储
mod=10**9+7
t,k=map(int,input().split())
prefix=[1]+[0]*(10**5)
for i in range(1,10**5+1):
    if i<k:
        prefix[i]=i+1
    else:
        last=1
        curr=last+prefix[i-k]
        prefix[i]=(prefix[i-1]+curr)%mod
        last=curr%mod

for _ in range(t):
    a,b=map(int,input().split())
    print((prefix[b]-prefix[a-1])%mod)

#构建 dp 数组和前缀和数组
mod=10**9+7
t,k=map(int,input().split())
dp=[1]+[0]*(10**5)
prefix=[0]+[0]*(10**5)
for i in range(1,10**5+1):
    if i<k:
        dp[i]=1
    else:
        dp[i]=(dp[i-1]+dp[i-k])
        dp[i]%mod
for i in range(1,10**5+1):
    prefix[i]=(prefix[i-1]+dp[i])%mod

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

代码运行截图 <mark>（至少包含有"Accepted"）4000KB 是代码 1，8200KB 是代码 2</mark>

474D - Flowers	Python 3	Accepted	624 ms	4000 KB
474D - Flowers	Python 3	Wrong answer on test 1	93 ms	4800 KB
474D - Flowers	Python 3	Wrong answer on test 3	93 ms	4000 KB
474D - Flowers	Python 3	Accepted	640 ms	8200 KB

LeetCode5.最长回文子串

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

思路:

我自己想到的是中心扩展的思路, 结果代码写的又臭又长且 debug 了半天。思路是遍历两次, 第一次是中心点是一个字符, 然后把每一个中心点对应的回文长度记录在字典 d1 里; 第二次是中心点是两个字符, 同理记录在 d2 里。由于可能不存在以 2 个字符为中心的, 所以要初始化 d2 为{0:0}防止后续取最大值的时候报错了。

看到了题解代码好简单, 主要的点是注意到中心点不管是 1 还是 2, 只要中心满足回文串, 只需要查看向两边扩散的两个字符是否相等即可, 这比我用切片和其 reverse 之后的比较要简洁; 同时只遍历一次, 每遍历一个就记录以该字符为 1 或 2 中心的最大长度, 然后使用 start 和 end 不断维护。好简洁。。。同时学习了 dp 的写法, 自己没想到, 题解应该有滑动窗口从小到大的感觉。

代码:

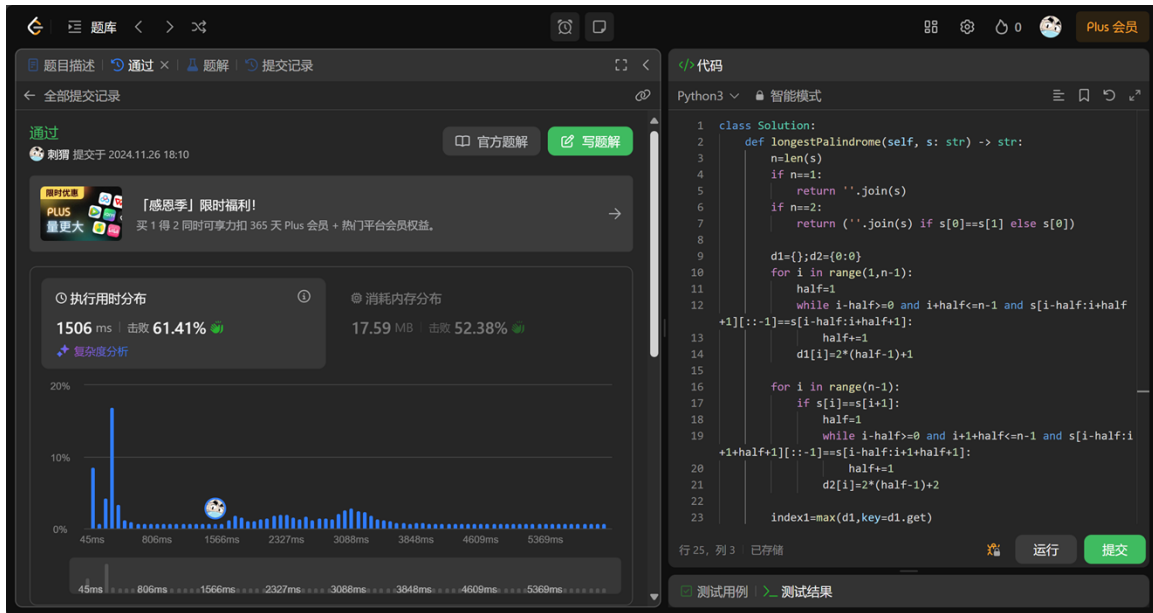
```
class Solution:
    def longestPalindrome(self, s: str) -> str:
        n=len(s)
        if n==1:
            return s
        if n==2:
            return (s if s[0]==s[1] else s[0])

        d1={};d2={0:0}
        for i in range(1,n-1):
            half=1
            while i-half>=0 and i+half<=n-1 and s[i-half:i+half+1][::-1]==s[i-half:i+half+1]:
                half+=1
            d1[i]=2*(half-1)+1

        for i in range(n-1):
            if s[i]==s[i+1]:
                half=1
                while i-half>=0 and i+1+half<=n-1 and s[i-half:i+1+half+1][::-1]==s[i-half:i+1+half+1]:
                    half+=1
                d2[i]=2*(half-1)+2

        index1=max(d1,key=d1.get)
        length1=d1[index1]
        index2=max(d2,key=d2.get)
        length2=d2[index2]
        if length1>length2:
            half=(length1-1)//2
            return s[index1-half:index1+half+1]
        half=(length2-2)//2
        return s[index2-half:index2+1+half+1]
```

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:

输入输出是 gpt 写的，自己写的一直 re；bfs 内容是看群里同学的代码和问 gpt 写的，后来一直 MLE；最后看老师的题解找不同找半天，发现一个更新的写法不同导致空间复杂度差距巨大，这我自己肯定是写不出来的。

代码:

```
from collections import deque
import sys
input=sys.stdin.read
dx,dy=[0,-1,1,0],[-1,0,0,1]
def bfs(s,x,y,i,j,water,m,n):
    q=deque([(x,y,s[x][y])])
    water[x][y]=s[x][y]
    while q:
        x,y,height=q.popleft()
        for _ in range(4):
            nx,ny=x+dx[_],y+dy[_]
            if 0<=nx<m and 0<=ny<n and s[nx][ny]<height and
water[nx][ny]<height:
                water[nx][ny]=height #####如果写成 water[nx][ny]=s[x][y]会 MLE
                q.append((nx, ny,height))
def main():
    data=input().split()
    idx = 0
    k = int(data[idx])
    idx += 1
    result = []
    for _ in range(k):
```

```

m, n = map(int, data[idx:idx+2])
idx += 2
#矩阵
s = [list(map(int, data[idx+i*n:idx+(i+1)*n])) for i in range(m)]
idx += m * n
#司令部
i, j = map(int, data[idx:idx+2])
i-=1;j-=1
idx += 2
p = int(data[idx])
idx += 1
water=[[0]*n for _ in range(m)]
#起点
for _ in range(p):
    x,y=map(int,data[idx:idx+2])
    idx+=2
    x-=1;y-=1
    if s[x][y]<=s[i][j]:
        continue
    bfs(s,x,y,i,j,water,m,n)
if water[i][j]>0:
    result.append('Yes')
else:
    result.append('No')
sys.stdout.write('\n'.join(result) + '\n')
if __name__ == '__main__':
    main()

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
from collections import deque
import sys
input=sys.stdin.read
dx,dy=[0,-1,1,0],[-1,0,0,1]
def bfs(s,x,y,i,j,water,m,n):
    q=deque([(x,y,s[x][y])])
    water[x][y]=s[x][y]
    while q:
        x,y,height=q.popleft()
        for _ in range(4):
            nx,ny=x+dx[_],y+dy[_]
            if 0<=nx<m and 0<=ny<n and s[nx][ny]<height and water[nx][ny]<0:
                water[nx][ny]=height
                q.append((nx, ny,height))
def main():
    data=input().split()
    idx = 0
    k = int(data[idx])
    idx += 1
    result = []
    for _ in range(k):
        m, n = map(int, data[idx:idx+2])
        idx += 2
        #矩阵
        s = [list(map(int, data[idx+i*n:idx+(i+1)*n])) for i in range(m)]
        idx += m * n
        #司令部
        i, j = map(int, data[idx:idx+2])
        i-=1;j-=1
        idx += 2
        p = int(data[idx])
        idx += 1
        water=[[0]*n for _ in range(m)]
        #起点
        for _ in range(p):
            x,y=map(int,data[idx:idx+2])
            idx+=2
            x-=1;y-=1
            if s[x][y]<=s[i][j]:
                continue
            bfs(s,x,y,i,j,water,m,n)
        if water[i][j]>0:
            result.append('Yes')
        else:
            result.append('No')
    sys.stdout.write('\n'.join(result) + '\n')
if __name__ == '__main__':
    main()
```

基本信息

#: 47443508
题目: 12029
提交人: 24n2400016635
内存: 6440kB
时间: 218ms
语言: Python3
提交时间: 2024-11-28 15:02:56

02802: 小游戏

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

思路:

本题笨人做了1天，最后对着答案改了1个小时。。。坑点:

①横纵坐标互换，读入n和m要换位置，x1,y1,x2,y2也要换位置

②因为终点不是空格，所以始终不可能保存到 **duque** 里面，因此在 **for** 循环内部要随时查看几个方向是否到达终点，如果到达就把线段添加的答案列表里面，再继续遍历其他方向，这也是为什么 **if s[nx][ny]==' '** 的判断语句后置的原因，同时也是因为这不是求最小步数

③所有的线段要全部保存，最后取最小值（但是我自己想不到）

④**deque** 保存的不仅仅是点的坐标，由于不能回溯所以要把当前这个点的方向状态和线段数状态以参数的形式向前传递

⑤最后要 **print()**

代码：

```
from collections import deque
dx,dy=[0,-1,1,0],[-1,0,0,1]
def bfs(x1,y1,seg,dir,segs):
    q=deque()
    q.append((x1,y1,seg,dir))
    inq=set()
    inq.add((x1,y1,dir))
    while q:
        for _ in range(len(q)):
            x,y,seg,dir=q.popleft()
            for i in range(4):
                nx,ny=x+dx[i],y+dy[i]
                if 0<=nx<n+2 and 0<=ny<m+2 and (nx,ny,i) not in inq:
                    new_seg=seg+(1 if dir!=i else 0)
                    new_dir=i
                    if (nx,ny)==(x2,y2):
                        segs.append(new_seg)
                        continue
                    if s[nx][ny]==' ':
                        q.append((nx,ny,new_seg,new_dir))
                        inq.add((nx,ny,i))
            if segs:
                return min(segs)
            else:
                return False

idx=0
while True:
    idx+=1
    m,n=map(int,input().split())
    if {m,n}=={0}:
        break
    s=[' '*(m+2)]+[' '+input()+' ' for _ in range(n)]+[' '*(m+2)]
    print(f'Board #{idx}:')
    a=[]
    i=0
    while True:
        i+=1
        y1,x1,y2,x2=map(int,input().split())
        if {x1,x2,y1,y2}=={0}:
            break
        ans=bfs(x1,y1,0,-1,[])
```



```
if ans:
    print(f'Pair {i}: {ans} segments.')
else:
    print(f'Pair {i}: impossible.')
print()
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47434218提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
from collections import deque
dx,dy=[0,-1,1,0],[-1,0,0,1]
def bfs(x1,y1,seg,dir,segs):
    q=deque()
    q.append((x1,y1,seg,dir))
    inq=set()
    inq.add((x1,y1,dir))
    while q:
        x,y,seg,dir=q.popleft()
        if (x,y)==(x2,y2):
            segs.append(seg)
            break
        for i in range(4):
            nx,ny=x+dx[i],y+dy[i]
            if 0<=nx<n+2 and 0<=ny<m+2 and (nx,ny,i) not in inq:
                new_seg=seg+(1 if dir!=i else 0)
                new_dir=i
                if (nx,ny)==(x2,y2):
                    segs.append(new_seg)
                    continue
                if s[nx][ny]==' ':
                    q.append((nx,ny,new_seg,new_dir))
                    inq.add((nx,ny,i))
    if segs:
        return min(segs)
    else:
        return False

idx=0
while True:
    idx+=1
    m,n=map(int,input().split())
    if {m,n}=={0}:
        break
    s=[' '* (m+2)] + [' '+input()+ ' ' for _ in range(n)] + [' '* (m+2)]
    print(f'Board #{idx}:')
    a=[]
    i=0
    while True:
        i+=1
        y1,x1,y2,x2=map(int,input().split())
        if {x1,x2,y1,y2}=={0}:
            break
        ans=bfs(x1,y1,0,-1,[])
        if ans:
            print(f'Pair {i}: {ans} segments.')
        else:
            print(f'Pair {i}: impossible.')
    print()
```

基本信息

#: 47434218
题目: 02802
提交人: 24n2400016635
内存: 4616kB
时间: 89ms
语言: Python3
提交时间: 2024-11-27 21:31:57

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

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>

回文子串数据是 10×3 ，因此完全可以二重循环，有可能就是要二维 dp 数组，感觉这个有引导作用？

Flowers 想到了数学方法用组合数去做，但是由于要用 factorial 函数，数据太大了导致时间复杂度不允许，不知道有没有优化方法

By twj_ink, contest: Codeforces Round 271 (Div. 2), problem: (D) Flowers, Time limit exceeded on test 1, #, [Copy](#)

```
from math import factorial

mod=10**9+7
def C(n,m):
    if n==m:
        return 1
    return factorial(n)//(factorial(m)*(n-m))

t,k=map(int,input().split())
dp=[0]*(10**5+1)
for i in range(1,k):
    dp[i]=i
for n in range(k,10**5+1):
    i=n//k
    d=n-i*k
    dp[n]=dp[n-1]%mod+1+sum(C(i+d+j*(k-1),i-j)%mod for j in range(i))
# print(dp[:5])

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

→Judgement Protocol

Test: #1, time: 1500 ms., memory: 9108 KB, exit code: -1, checker exit code: 0, verdict: TIME_LIMIT_EXCEEDED

Input

```
3 2
1 3
2 3
4 4
```

后两题好难啊而且是模板变形，花了我 1 天又 3/4 左右的时间吧，群里同学怎么做到的几个小时啊😓

你的提交记录

#	结果	时间
51	Accepted	2024-11-28
50	Accepted	2024-11-28
49	Accepted	2024-11-28
48	Memory Limit Exceeded	2024-11-28
47	Compile Error	2024-11-28
46	Memory Limit Exceeded	2024-11-28
45	Memory Limit Exceeded	2024-11-28
44	Wrong Answer	2024-11-28
43	Memory Limit Exceeded	2024-11-28
42	Wrong Answer	2024-11-28
41	Runtime Error	2024-11-28
40	Runtime Error	2024-11-28
39	Runtime Error	2024-11-28
38	Wrong Answer	2024-11-28
37	Wrong Answer	2024-11-28
36	Wrong Answer	2024-11-28
35	Wrong Answer	2024-11-28
34	Wrong Answer	2024-11-28
33	Wrong Answer	2024-11-28
32	Time Limit Exceeded	2024-11-28
31	Time Limit Exceeded	2024-11-28
30	Wrong Answer	2024-11-28
29	Wrong Answer	2024-11-28
28	Wrong Answer	2024-11-28
27	Wrong Answer	2024-11-28
26	Wrong Answer	2024-11-28
25	Runtime Error	2024-11-28
24	Runtime Error	2024-11-28
23	Runtime Error	2024-11-28
22	Wrong Answer	2024-11-27
21	Runtime Error	2024-11-27
20	Runtime Error	2024-11-27
19	Runtime Error	2024-11-27
18	Wrong Answer	2024-11-27
17	Runtime Error	2024-11-27
16	Wrong Answer	2024-11-27
15	Runtime Error	2024-11-27
14	Runtime Error	2024-11-27
13	Runtime Error	2024-11-27
12	Runtime Error	2024-11-27
11	Runtime Error	2024-11-27
10	Runtime Error	2024-11-27
9	Runtime Error	2024-11-27
8	Runtime Error	2024-11-27
7	Compile Error	2024-11-27
6	Runtime Error	2024-11-27
5	Runtime Error	2024-11-27
4	Runtime Error	2024-11-27
3	Runtime Error	2024-11-27
2	Runtime Error	2024-11-27
1	Compile Error	2024-11-27