1. (6min)

LuoguP1255 数楼梯

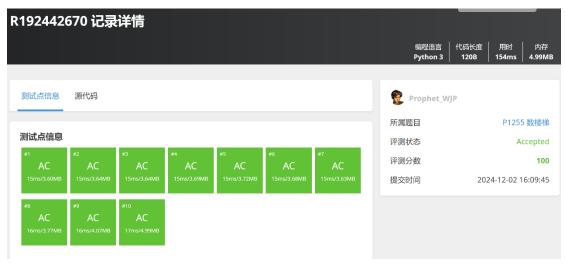
dp, bfs, https://www.luogu.com.cn/problem/P1255

思路:显然为斐波那契数列

代码:

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

运行:



2. (5min)

27528: 跳台阶

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

思路: 第1到 n-1级台阶, 每个台阶都可以选择经过或者不经

过, 共 2^(n-1)种

代码:

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

运行:



3. (20min)

474D. Flowers

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

思路: dp, 递推即可

代码:

```
N=1e9+7
t, k=map(int,input().split())
da=[[0,0] for _ in range(t)]
ma=0
for i in range(t):
    da[i][0],da[i][1]=map(int,input().split())
    ma=max(ma,da[i][1])
dp=[1 for x in range(ma+1)]
num=[x for x in range(ma+1)]
for s in range(k,ma+1):
    dp[s]=(dp[s-1]+dp[s-k])%N
    num[s]=(num[s-1]+dp[s])%N
for p in range(t):
    print(int((num[da[p][1]]-num[da[p][0]-1])%N))
```

运行:



4. (25min)

LeetCode5.最长回文子串

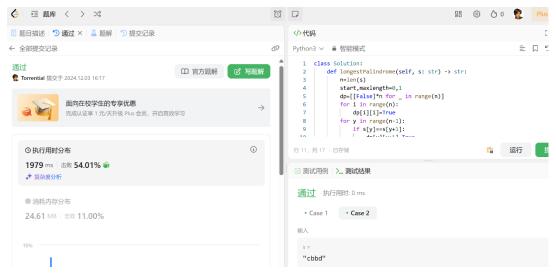
dp, two pointers, string, https://leetcode.cn/problems/longest-palindromic-substring/

思路:以字串的长度为指标从小到大 dp

代码:

```
class Solution:
   def longestPalindrome(self, s: str) -> str:
      n=len(s)
      start, maxlength=0,1
      dp=[[False]*n for    in range(n)]
      for i in range(n):
          dp[i][i]=True
       for y in range (n-1):
          if s[y] == s[y+1]:
              dp[y][y+1]=True
              start, maxlength=y, 2
       for x in range (2, n):
          for z in range (n-x):
              if s[z] == s[z+x] and dp[z+1][z+x-1]:
                 dp[z][z+x]=True
                 start, maxlength=z, x+1
      return(s[start:start+maxlength])
```

运行:



5. (2h)

12029: 水淹七军

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

思路: 常规的 dfs, 但是 debug 搞了半天……最后发现是 break 的问题

代码:

```
from collections import deque
import sys
def bfs(x, y):
   global sl1, sl2, mat
   dir = [[0, -1], [0, 1], [1, 0], [-1, 0]]
   q=deque([(x,y)])
   while q:
       e,f=q.popleft()
       for dx, dy in dir:
          nx, ny=e+dx, f+dy
          if mat[e][f]>mat[nx][ny]:
              q.append((nx,ny))
              if nx==sl1 and ny==sl2:
                 return True
              mat[nx][ny]=mat[e][f]
   return False
lines=list(sys.stdin.read().split())
K=int(lines[0])
```

```
for in range(K):
   m, n=int(lines[i]), int(lines[i+1])
   mat = [[1025] * (n+2)]
   for a in range (m):
mat.append([1025]+list(map(int,lines[i:i+n]))+[1025
])
       i+=n
   mat.append([1025]*(n+2))
   sl1, sl2=int(lines[i]), int(lines[i+1])
   i+=2
   waternum=int(lines[i])
   answer=False
   for b in range(waternum):
       if bfs(int(lines[i]),int(lines[i+1])):
          answer=True
       i=i+2
   if answer:
   else:
```

运行:



状态: Accepted

```
源代码
 from collections import deque
 import sys
     global sl1,sl2,mat
dir = [[0, -1], [0, 1], [1, 0], [-1, 0]]
q-deque([(x,y)])
      while q:
          e,f=q.popleft()
              nx, ny=e+dx, f+dy
               if mat[e][f]>mat[nx][ny]:
                   q.append((nx,ny))
                   if nx--sl1 and ny--sl2:
                       return True
                   mat[nx][ny]=mat[e][f]
      return False
 lines=list(sys.stdin.read().split())
 for in range(K):
      m, n=int(lines[i]), int(lines[i+1])
```

#: 47538729 题目: 12029 提交人: 24n2400011028 内存: 5908kB 时间: 127ms 语言: Python3

基本信息

提交时间: 2024-12-03 19:21:58

6. (1.5h)

02802: 小游戏

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

思路: dfs 时用指标标识上一次的方向,并且用 history 记录以某个特定方向到达某个点的最短线段数

代码:

```
from collections import deque
def bfs (x1, y1, x2, y2):
   global step
   minstep=6400
   dir=[[0,-1],[0,1],[1,0],[-1,0]]
   history={}
   q=deque([(x1,y1,-1,0)])
   while q:
       x,y,di,num= q.popleft()
       for i in range (4):
          dx, dy=dir[i][0], dir[i][1]
          nx, ny=x+dx, y+dy
          if i==di:
              newnum=num
          else:
              newnum=num+1
          if nx==x2 and ny==y2:
              minstep=min(minstep, newnum)
          if 0 \le nx \le h+1 and 0 \le ny \le w+1 and
mat[nx][ny] == 0 and ((nx, ny, i) not in history or
history[(nx,ny,i)]>newnum):
                 q.append((nx,ny,i,newnum))
                 history[(nx, ny, i)] = newnum
   if minstep<6400:
       return(str(minstep)+' segments.')
   else:
n=1
while True:
  w,h=map(int,input().split())
```

```
matstring=[]
   if w==h==0:
      break
   for in range(h):
      matstring.append(input())
   mat = [[0]*(w+2) for in range(h+2)]
   for i in range(h):
      for j in range(w):
          if matstring[i][j]=='X':
             mat[i+1][j+1]=1
   ans='Board #'+str(n)+':\n'
   n+=1
  m=1
   while True:
      x11, y11, x22, y22=map(int, input().split())
      if x11==y11==x22==y22==0:
         break
      ans+= 'Pair '+str (m) + ':
'+bfs(y11,x11,y22,x22)+'\n'
      m+=1
   print(ans)
```

运行:



基本信息

#: 47542709 题目: 02802

内存: 4740kB

语言: Python3

时间: 85ms

提交人: 24n2400011028

提交时间: 2024-12-03 22:44:21

#47542709提交状态 查看 提交 统计 提问

状态: Accepted

```
源代码
 from collections import deque
 def bfs(x1,y1,x2,y2):
    global step
    dir=[[0,-1],[0,1],[1,0],[-1,0]]
    history-{}
    q=deque([(x1,y1,-1,0)])
    while q:
        x,y,di,num= q.popleft()
        for i in range(4):
           dx, dy=dir[i][0], dir[i][1]
           nx, ny=x+dx, y+dy
           if i==di:
              newnum=num
              newnum=num+1
           if nx==x2 and ny==y2:
              minstep=min(minstep,newnum)
           history[(nx, ny, i)] = newnum
    if minstep<6400:
```

总结与收获:

- 1. 巩固了 dfs 的栈写法, 感觉这个方法似乎在某些方面更加普适一些, 而且比较方便 return 回想要的值?
- 2. 学到了一些特殊的输入格式写法