Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by 邱泽霖 化学与分子工程学院

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

18160: 最大连通域面积

dfs similar, http://cs101.openjudge.cn/practice/18160

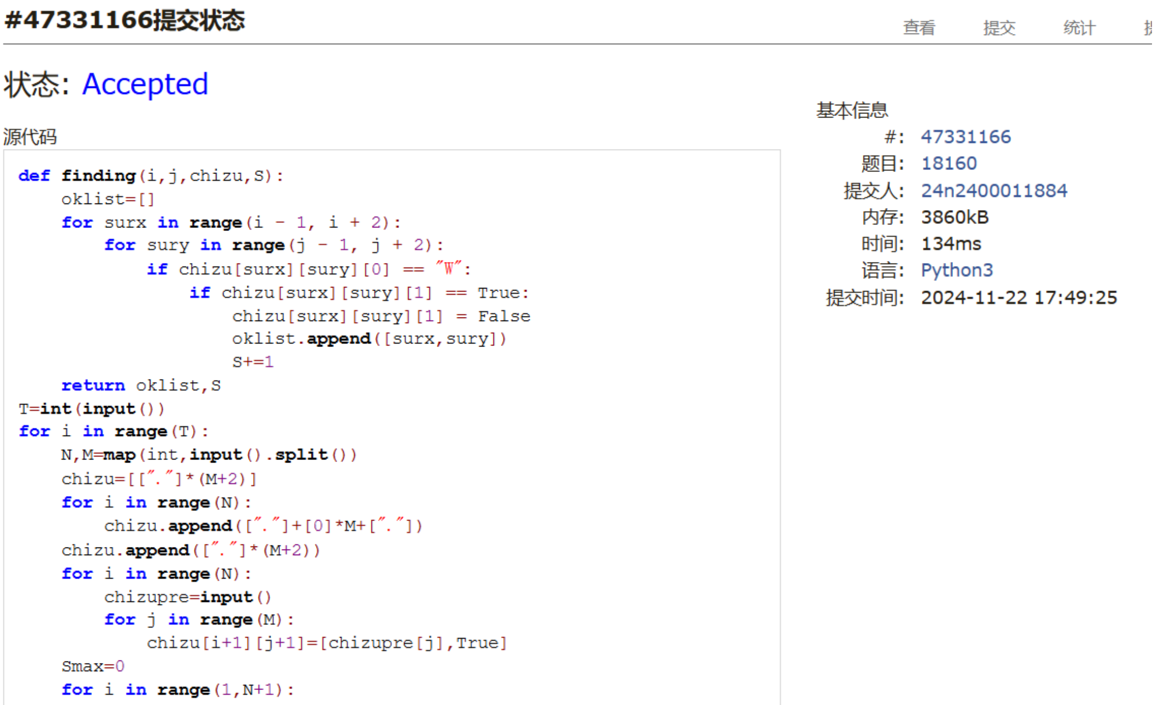
思路：

每遇到一个W，就把与它相连的W全都数出来。用时约30min

代码：

```
def finding(i,j,chizu,S):
    oklist=[]
    for surx in range(i - 1, i + 2):
        for sury in range(j - 1, j + 2):
            if chizu[surx][sury][0] == "W":
                if chizu[surx][sury][1] == True:
                    chizu[surx][sury][1] = False
                    oklist.append([surx,sury])
                    S+=1
    return oklist,S
T=int(input())
for i in range(T):
    N,M=map(int,input().split())
    chizu=[["."]*(M+2)]
    for i in range(N):
        chizu.append(["."]+[0]*M+["."])
    chizu.append(["."]*(M+2))
    for i in range(N):
        chizupre=input()
        for j in range(M):
            chizu[i+1][j+1]=[chizupre[j],True]
    Smax=0
    for i in range(1,N+1):
        for j in range(1,M+1):
            if chizu[i][j][0]=="W":
                if chizu[i][j][1]==True:
                    chizu[i][j][1]=False
                    S=1
                    Smax=max(Smax,S)
                    oklist,S=finding(i,j,chizu,S)
                    while True:
                        ok=True
                        nnoklist=[]
                        for k in oklist:
                            noklist,S=finding(k[0],k[1],chizu,S)
                            nnoklist+=noklist
                        if len(nnoklist)==0:
                            Smax=max(Smax,S)
                            break
                        else:
                            oklist=nnoklist
    print(Smax)
```

代码运行截图 （至少包含有"Accepted"）



#47331166提交状态

状态: Accepted

源代码

```
def finding(i,j,chizu,S):
    oklist=[]
    for surx in range(i - 1, i + 2):
        for sury in range(j - 1, j + 2):
            if chizu[surx][sury][0] == "W":
                if chizu[surx][sury][1] == True:
                    chizu[surx][sury][1] = False
                    oklist.append([surx,sury])
                    S+=1
    return oklist,S
T=int(input())
for i in range(T):
    N,M=map(int,input().split())
    chizu=[["."]*(M+2)]
    for i in range(N):
        chizu.append(["."]+[0]*M+["."])
    chizu.append(["."]*(M+2))
    for i in range(N):
        chizupre=input()
        for j in range(M):
            chizu[i+1][j+1]=[chizupre[j],True]
    Smax=0
    for i in range(1,N+1):
```

基本信息

#:	47331166
题目:	18160
提交人:	24n2400011884
内存:	3860kB
时间:	134ms
语言:	Python3
提交时间:	2024-11-22 17:49:25

19930: 寻宝

bfs, http://cs101.openjudge.cn/practice/19930

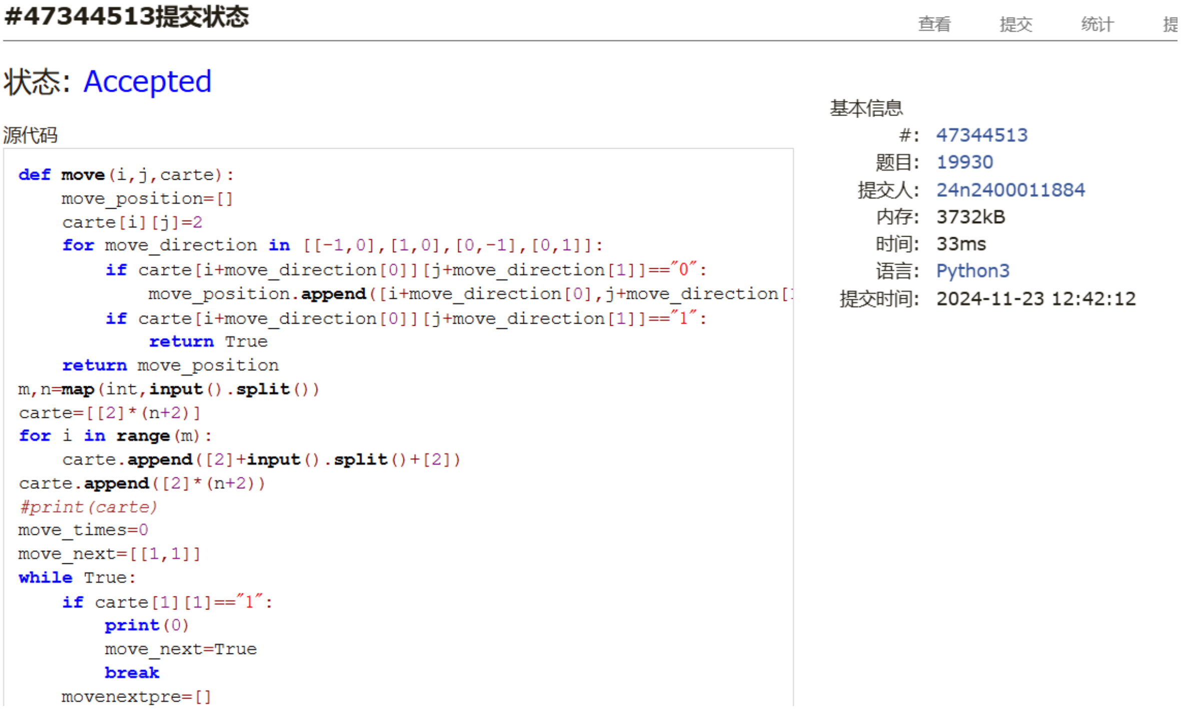
思路：

每次尝试朝未走过的格子走，记录走过的步数，遇到1时输出。用时约30min

代码：

```
def move(i,j,carte):
    move_position=[]
    carte[i][j]=2
    for move_direction in [[-1,0],[1,0],[0,-1],[0,1]]:
        if carte[i+move_direction[0]][j+move_direction[1]]=="0":
            move_position.append([i+move_direction[0],j+move_direction[1]])
        if carte[i+move_direction[0]][j+move_direction[1]]=="1":
            return True
    return move_position
m,n=map(int,input().split())
carte=[[2]*(n+2)]
for i in range(m):
    carte.append([2]+input().split()+[2])
carte.append([2]*(n+2))
#print(carte)
move_times=0
move_next=[[1,1]]
while True:
    if carte[1][1]=="1":
        print(0)
        move_next=True
        break
    movenextpre=[]
    if len(move_next)==0:
        move_next=False
        break
    for i in move_next:
        #print(i)
        move_next_pre=move(i[0],i[1],carte)
        #if move_next_pre:
            #print(move_next_pre)
        if move_next_pre!=True:
            movenextpre+=move_next_pre
        else:
            move_next=True
            break
        #print(move_next_pre)
    move_times+=1
    if move_next==True:
        print(move_times)
        break
    else:
        move_next=movenextpre
if not move_next:
    print("NO")
```

代码运行截图 == （至少包含有"Accepted"） ==



#47344513提交状态

状态: Accepted

源代码

```
def move(i,j,carte):
    move_position=[]
    carte[i][j]=2
    for move_direction in [[-1,0],[1,0],[0,-1],[0,1]]:
        if carte[i+move_direction[0]][j+move_direction[1]]=="0":
            move_position.append([i+move_direction[0],j+move_direction[
        if carte[i+move_direction[0]][j+move_direction[1]]=="1":
            return True
    return move_position
m,n=map(int,input().split())
carte=[[2]*(n+2)]
for i in range(m):
    carte.append([2]+input().split()+[2])
carte.append([2]*(n+2))
#print(carte)
move_times=0
move_next=[[1,1]]
while True:
    if carte[1][1]=="1":
        print(0)
        move_next=True
        break
    movenextpre=[]
```

基本信息

#: 47344513
题目: 19930
提交人: 24n2400011884
内存: 3732kB
时间: 33ms
语言: Python3
提交时间: 2024-11-23 12:42:12

04123: 马走日

dfs, http://cs101.openjudge.cn/practice/04123

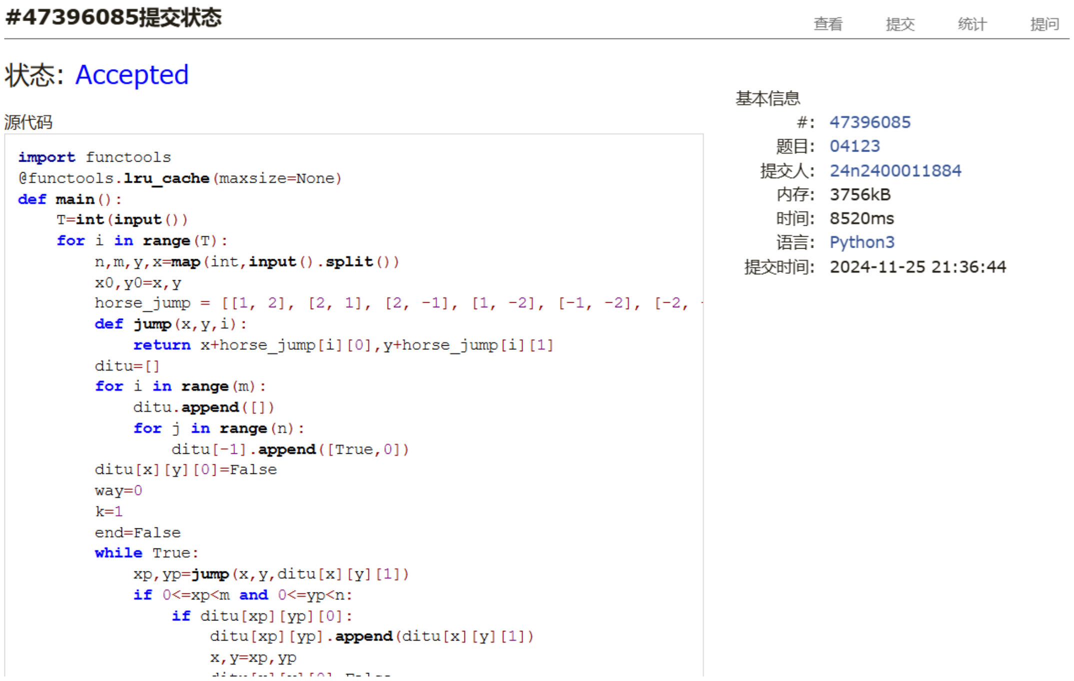
思路:

朝8个方向跳，跳出边界或重复时转向。最初超时，用了lru_cache后AC。但尝试了6 6 0 0的数据，仍不能在正常时间内给出答案。用时约90min

代码:

```
import functools
@functools.lru_cache(maxsize=None)
def main():
    T=int(input())
    for i in range(T):
        n,m,y,x=map(int,input().split())
        x0,y0=x,y
        horse_jump = [[1, 2], [2, 1], [2, -1], [1, -2], [-1, -2], [-2, -1], [-2, 1], [-1, 2]]
        def jump(x,y,i):
            return x+horse_jump[i][0],y+horse_jump[i][1]
        ditu=[]
        for i in range(m):
            ditu.append([])
            for j in range(n):
                ditu[-1].append([True,0])
        ditu[x][y][0]=False
        way=0
        k=1
        end=False
        while True:
            xp,yp=jump(x,y,ditu[x][y][1])
            if 0<=xp<m and 0<=yp<n:
                if ditu[xp][yp][0]:
                    ditu[xp][yp].append(ditu[x][y][1])
                    x,y=xp,yp
                    ditu[x][y][0]=False
                    k+=1
                else:
                    while True:
                        if ditu[x][y][1]<7:
                            ditu[x][y][1]+=1
                            break
                        elif x==x0 and y==y0:
                            end=True
                            break
                        else:
                            xp,yp=x,y
                            x,y=jump(x,y,ditu[x][y][2]-4)
                            ditu[xp][yp]=[True,0]
                            k-=1
            else:
                while True:
                    if ditu[x][y][1] < 7:
                        ditu[x][y][1] += 1
                        break
                    elif x==x0 and y==y0:
                        end=True
                        break
                    else:
                        xp,yp=x,y
                        x, y = jump(x, y, ditu[x][y][2] - 4)
                        ditu[xp][yp]=[True,0]
                        k -= 1
            if k==m*n:
                way+=1
            if end:
                break
        print(way//8)
main()
```

代码运行截图 （至少包含有"Accepted"）



#47396085提交状态

状态: Accepted

源代码

```
import functools
@functools.lru_cache(maxsize=None)
def main():
    T=int(input())
    for i in range(T):
        n,m,y,x=map(int,input().split())
        x0,y0=x,y
        horse_jump = [[1, 2], [2, 1], [2, -1], [1, -2], [-1, -2], [-2,
        def jump(x,y,i):
            return x+horse_jump[i][0],y+horse_jump[i][1]
        ditu=[]
        for i in range(m):
            ditu.append([])
            for j in range(n):
                ditu[-1].append([True,0])
        ditu[x][y][0]=False
        way=0
        k=1
        end=False
        while True:
            xp,yp=jump(x,y,ditu[x][y][1])
            if 0<=xp<m and 0<=yp<n:
                if ditu[xp][yp][0]:
                    ditu[xp][yp].append(ditu[x][y][1])
                    x,y=xp,yp
                    ditu[x][y][0]=False
```

基本信息

#:	47396085
题目:	04123
提交人:	24n2400011884
内存:	3756kB
时间:	8520ms
语言:	Python3
提交时间:	2024-11-25 21:36:44

sy316: 矩阵最大权值路径

dfs, https://sunnywhy.com/sfbj/8/1/316

思路:

深度优先搜索，每条路走到底后回退，用时约20min

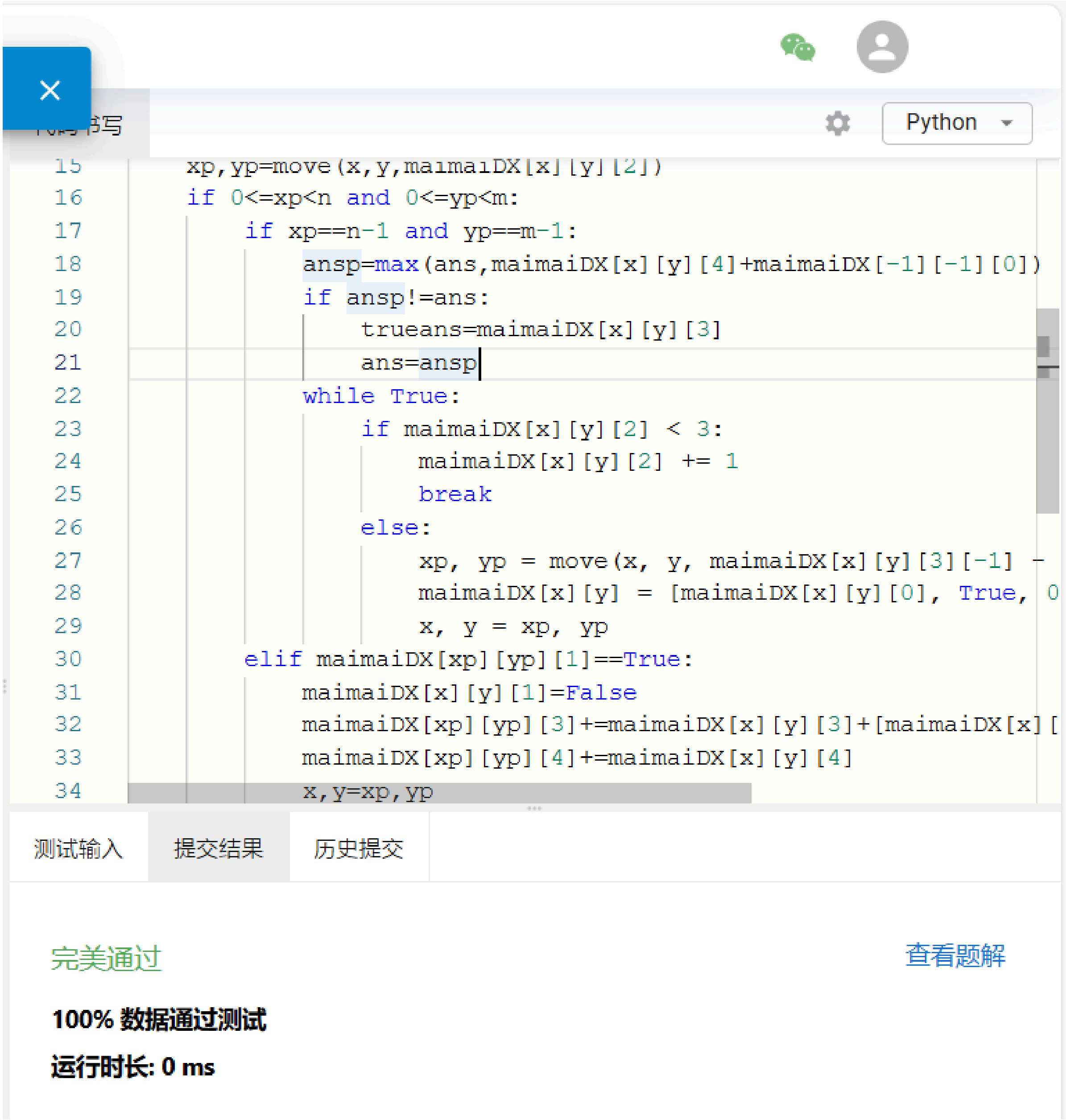
代码:

```
n,m=map(int,input().split())
movedir = [[0, 1], [1, 0], [0, -1], [-1, 0]]
def move(x,y,i):
    return x+movedir[i][0], y+movedir[i][1]

x=y=0
maimaiDX=[]
for i in range(n):
    maimaiDX.append(list(map(int,input().split())))
    for j in range(m):
        maimaiDX[-1][j]=[maimaiDX[-1][j],True,0,[],maimaiDX[-1][j]]
maimaiDX[-1][-1][1]=False
ans=-1E9
while True:
    xp,yp=move(x,y,maimaiDX[x][y][2])
    if 0<=xp<n and 0<=yp<m:
        if xp==n-1 and yp==m-1:
            ansp=max(ans,maimaiDX[x][y][4]+maimaiDX[-1][-1][0])
            if ansp!=ans:
                trueans=maimaiDX[x][y][3]
                ans=ansp
            while True:
                if maimaiDX[x][y][2] < 3:
                    maimaiDX[x][y][2] += 1
                    break
                else:
                    xp, yp = move(x, y, maimaiDX[x][y][3][-1] - 2)
                    maimaiDX[x][y] = [maimaiDX[x][y][0], True, 0, [], maimaiDX[x][y][0]]
                    x, y = xp, yp
        elif maimaiDX[xp][yp][1]==True:
            maimaiDX[x][y][1]=False
            maimaiDX[xp][yp][3]+=maimaiDX[x][y][3]+[maimaiDX[x][y][2]]
            maimaiDX[xp][yp][4]+=maimaiDX[x][y][4]
            x,y=xp,yp
        else:
            while True:
                if maimaiDX[x][y][2]<3:
                    maimaiDX[x][y][2]+=1
                    break
                else:
                    xp,yp=move(x,y,maimaiDX[x][y][3][-1]-2)
                    maimaiDX[x][y]=[maimaiDX[x][y][0],True,0,[],maimaiDX[x][y][0]]
                    x,y=xp,yp
    else:
        while True:
            if maimaiDX[x][y][2] < 3:
                maimaiDX[x][y][2] += 1
                break
            else:
                xp, yp = move(x, y, maimaiDX[x][y][3][-1] - 2)
                maimaiDX[x][y] = [maimaiDX[x][y][0], True, 0, [], maimaiDX[x][y][0]]
                x, y = xp, yp
    if maimaiDX[0][0][2]>=2:
        break

x=y=0
for i in trueans:
    print(x+1,y+1)
    x,y=move(x,y,i)
print(x+1,y+1)
print(n,m)
```

代码运行截图 （至少包含有"Accepted"）



```
        xp,yp=move(x,y,maimaiDX[x][y][2])
        if 0<=xp<n and 0<=yp<m:
            if xp==n-1 and yp==m-1:
                ansp=max(ans,maimaiDX[x][y][4]+maimaiDX[-1][-1][0])
                if ansp!=ans:
                    trueans=maimaiDX[x][y][3]
                    ans=ansp
                while True:
                    if maimaiDX[x][y][2] < 3:
                        maimaiDX[x][y][2] += 1
                        break
                    else:
                        xp, yp = move(x, y, maimaiDX[x][y][3][-1] -
                        maimaiDX[x][y] = [maimaiDX[x][y][0], True, 0
                        x, y = xp, yp
            elif maimaiDX[xp][yp][1]==True:
                maimaiDX[x][y][1]=False
                maimaiDX[xp][yp][3]+=maimaiDX[x][y][3]+[maimaiDX[x][
                maimaiDX[xp][yp][4]+=maimaiDX[x][y][4]
                x,y=xp,yp
```

完美通过

100% 数据通过测试

运行时长: 0 ms

LeetCode62.不同路径

dp, https://leetcode.cn/problems/unique-paths/

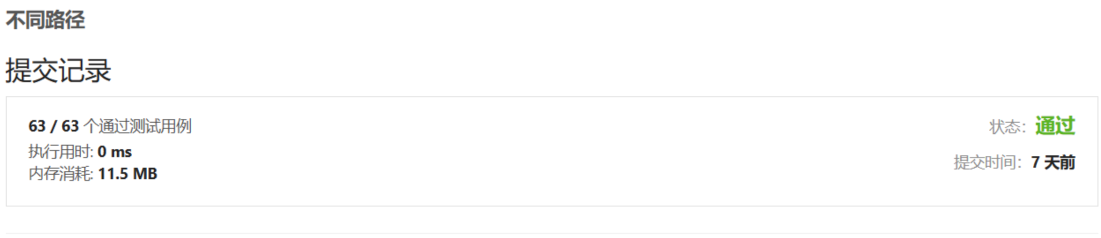
思路:

使用高中的组合数知识加以解决，用时3min

代码:

```
class Solution(object):
    def uniquePaths(self, m, n):
        import math
        return math.factorial(m+n-2)/math.factorial(m-1)/math.factorial(n-1)
```

代码运行截图 （至少包含有"Accepted"）



不同路径

提交记录

63 / 63 个通过测试用例

执行用时: 0 ms

内存消耗: 11.5 MB

状态: 通过

提交时间: 7 天前

sy358: 受到祝福的平方

dfs, dp, https://sunnywhy.com/sfbj/8/3/539

思路：

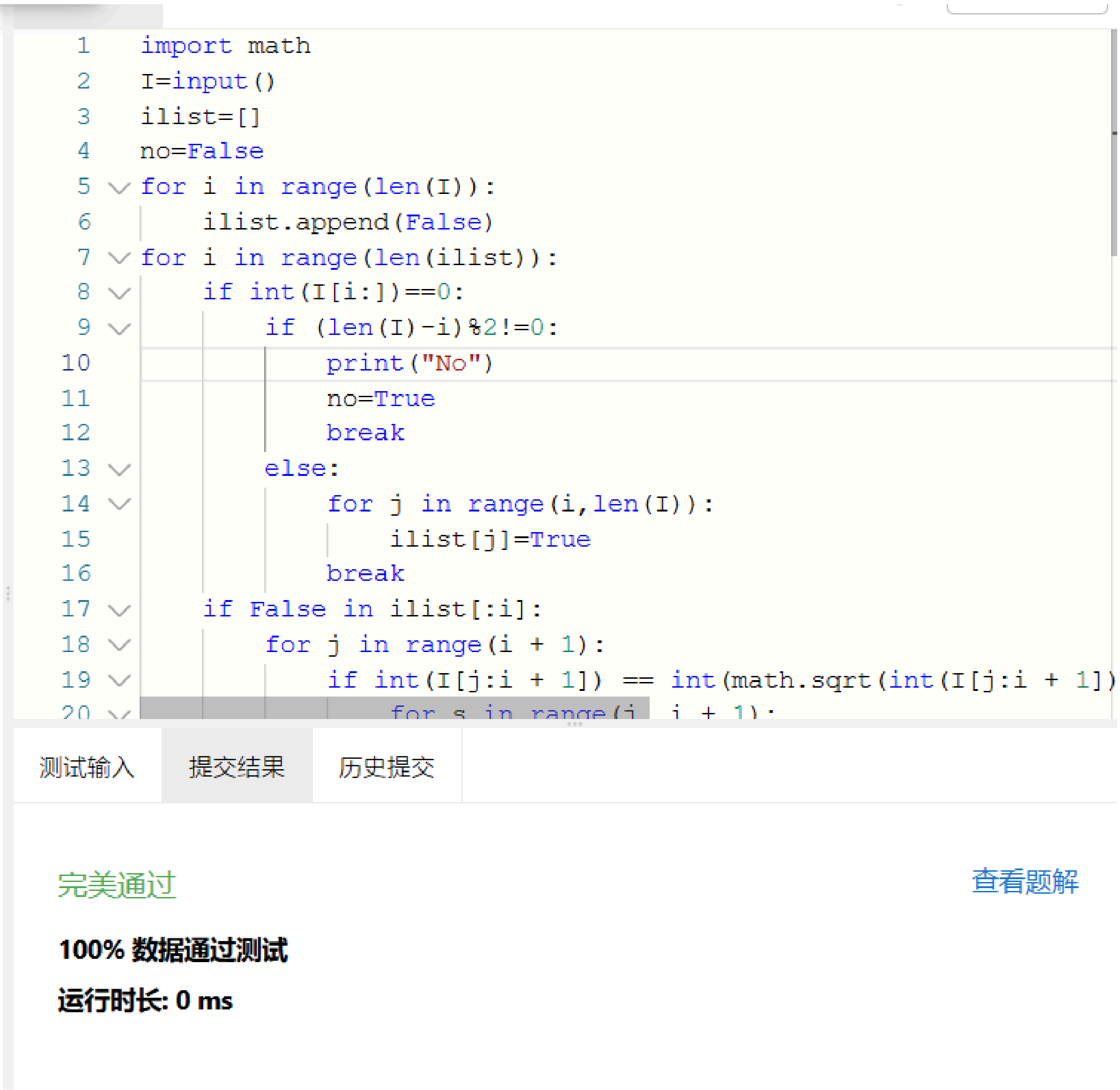
标记每一个平方数的开头。若该平方数可与后面的数组成更大的平方数，且后面的数无法自己组成平方数，则替换为该平方数。用时1h

代码：

```
import math
I=input()
ilist=[]
no=False
for i in range(len(I)):
    ilist.append(False)
for i in range(len(ilist)):
    if int(I[i:])==0:
        if (len(I)-i)%2!=0:
            print("No")
            no=True
            break
        else:
            for j in range(i,len(I)):
                ilist[j]=True
            break
    if False in ilist[:i]:
        for j in range(i + 1):
            if int(I[j:i + 1]) == int(math.sqrt(int(I[j:i + 1]))) ** 2 and
(ilist[j] == "head" or ilist[j] == False):
                for s in range(j, i + 1):
                    ilist[s] = True
                ilist[j] = "head"
                if ilist[j] == "0":
                    for k in range(j + 1, i + 1):
                        if I[k] == "0":
                            ilist[k]="head"
                        else:
                            ilist[k]="head"
                            break
                break
    elif int(I[i]) in [0,1,4,9]:
        ilist[i]="head"
    else:
        for j in range(i + 1):
            if int(I[j:i + 1]) == int(math.sqrt(int(I[j:i + 1]))) ** 2 and (ilist[j] == "head" or ilist[j] == False):
                for s in range(j, i + 1):
                    ilist[s] = True
                ilist[j] = "head"
                if ilist[j] == "0":
                    for k in range(j + 1, i + 1):
                        if I[k] == "0":
                            ilist[k]="head"
                        else:
                            ilist[k]="head"
                            break
                break

if not no:
    for k in range(len(ilist)):
        if ilist[k] == False:
            print("No")
            no = True
            break
    if not no:
        print("Yes")
```

代码运行截图 （至少包含有"Accepted"）



```python
import math
I=input()
ilist=[]
no=False
for i in range(len(I)):
    ilist.append(False)
for i in range(len(ilist)):
    if int(I[i:])==0:
        if (len(I)-i)%2!=0:
            print("No")
            no=True
            break
        else:
            for j in range(i,len(I)):
                ilist[j]=True
            break
    if False in ilist[:i]:
        for j in range(i + 1):
            if int(I[j:i + 1]) == int(math.sqrt(int(I[j:i + 1]))
                for s in range(i   i + 1):
```

完美通过

100% 数据通过测试

运行时长: 0 ms

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。

这周一直在生病，所以只完成了作业题目。dfs与bfs写法并不困难，都不看题解自己写出来了，但花费时间往往较长，需要看题解加以改进。收到祝福的平方debug了好久，总会忽略一些情况。希望下周多做题可以改善这种现象。GitHub连接不稳定，想看题解和选做题时有时需要链接好久。不知道有没有什么不挂梯子的解决办法。