Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by 俞天麒 物理学院

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

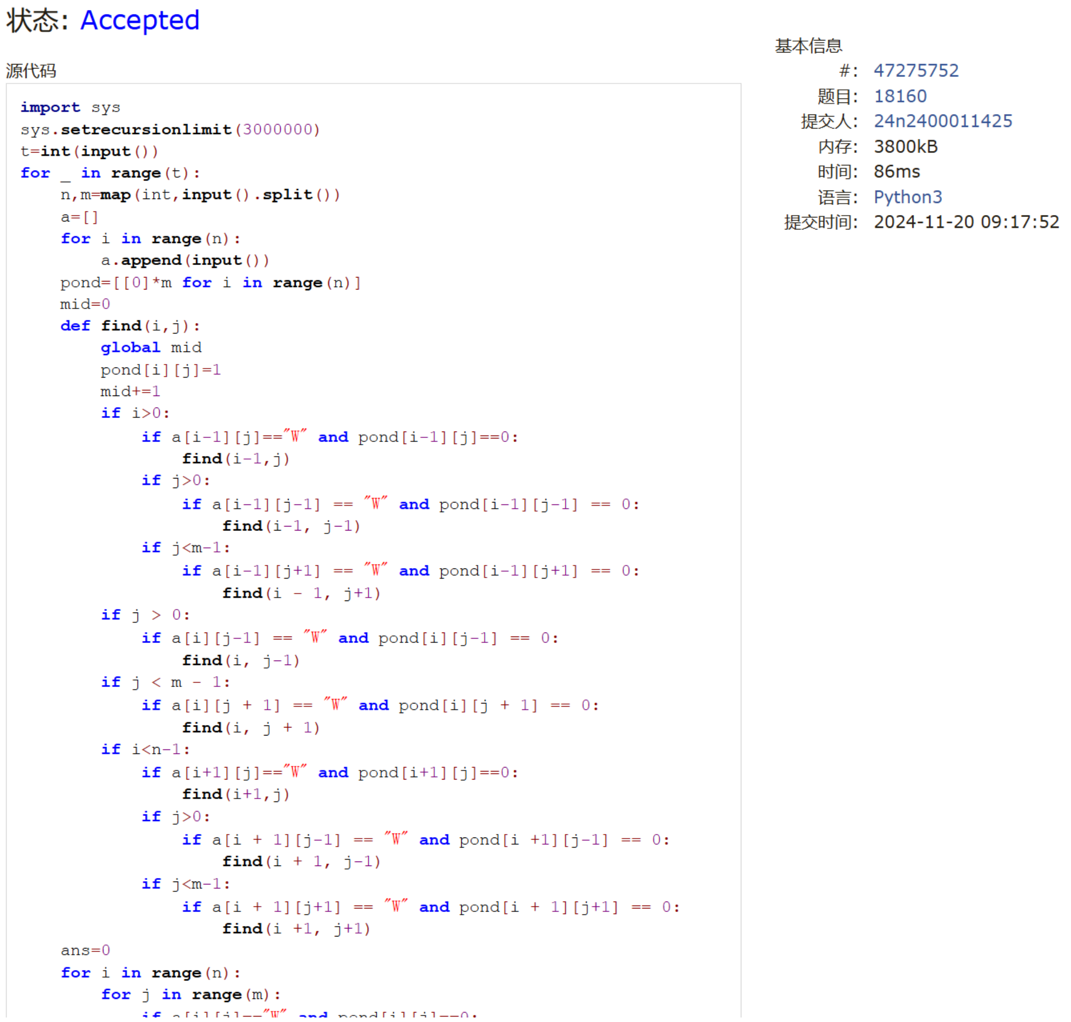
思路：

找到一个没被标记的点，然后从这里出发，用递归的方式遍历周围的8个点，每遍历一个点面积+1。我这里好像对边界的处理有点复杂了。

代码：

```python
import sys
sys.setrecursionlimit(3000000)
t=int(input())
for _ in range(t):
    n,m=map(int,input().split())
    a=[]
    for i in range(n):
        a.append(input())
    pond=[[0]*m for i in range(n)]
    mid=0
    def find(i,j):
        global mid
        pond[i][j]=1
        mid+=1
        if i>0:
            if a[i-1][j]=="W" and pond[i-1][j]==0:
                find(i-1,j)
            if j>0:
                if a[i-1][j-1] == "W" and pond[i-1][j-1] == 0:
                    find(i-1, j-1)
            if j<m-1:
                if a[i-1][j+1] == "W" and pond[i-1][j+1] == 0:
                    find(i - 1, j+1)
        if j > 0:
            if a[i][j-1] == "W" and pond[i][j-1] == 0:
                find(i, j-1)
        if j < m - 1:
            if a[i][j + 1] == "W" and pond[i][j + 1] == 0:
                find(i, j + 1)
        if i<n-1:
            if a[i+1][j]=="W" and pond[i+1][j]==0:
                find(i+1,j)
            if j>0:
                if a[i + 1][j-1] == "W" and pond[i +1][j-1] == 0:
                    find(i + 1, j-1)
            if j<m-1:
                if a[i + 1][j+1] == "W" and pond[i + 1][j+1] == 0:
                    find(i +1, j+1)
    ans=0
    for i in range(n):
        for j in range(m):
            if a[i][j]=="W" and pond[i][j]==0:
                mid=0
                find(i,j)
                ans=max(ans,mid)
    print(ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
import sys
sys.setrecursionlimit(3000000)
t=int(input())
for _ in range(t):
    n,m=map(int,input().split())
    a=[]
    for i in range(n):
        a.append(input())
    pond=[[0]*m for i in range(n)]
    mid=0
    def find(i,j):
        global mid
        pond[i][j]=1
        mid+=1
        if i>0:
            if a[i-1][j]=="W" and pond[i-1][j]==0:
                find(i-1,j)
            if j>0:
                if a[i-1][j-1] == "W" and pond[i-1][j-1] == 0:
                    find(i-1, j-1)
            if j<m-1:
                if a[i-1][j+1] == "W" and pond[i-1][j+1] == 0:
                    find(i - 1, j+1)
        if j > 0:
            if a[i][j-1] == "W" and pond[i][j-1] == 0:
                find(i, j-1)
        if j < m - 1:
            if a[i][j + 1] == "W" and pond[i][j + 1] == 0:
                find(i, j + 1)
        if i<n-1:
            if a[i+1][j]=="W" and pond[i+1][j]==0:
                find(i+1,j)
            if j>0:
                if a[i + 1][j-1] == "W" and pond[i +1][j-1] == 0:
                    find(i + 1, j-1)
            if j<m-1:
                if a[i + 1][j+1] == "W" and pond[i + 1][j+1] == 0:
                    find(i +1, j+1)
    ans=0
    for i in range(n):
        for j in range(m):
            if a[i][j]=="W" and pond[i][j]==0:
```

基本信息

#:	47275752
题目:	18160
提交人:	24n2400011425
内存:	3800kB
时间:	86ms
语言:	Python3
提交时间:	2024-11-20 09:17:52

19930: 寻宝

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

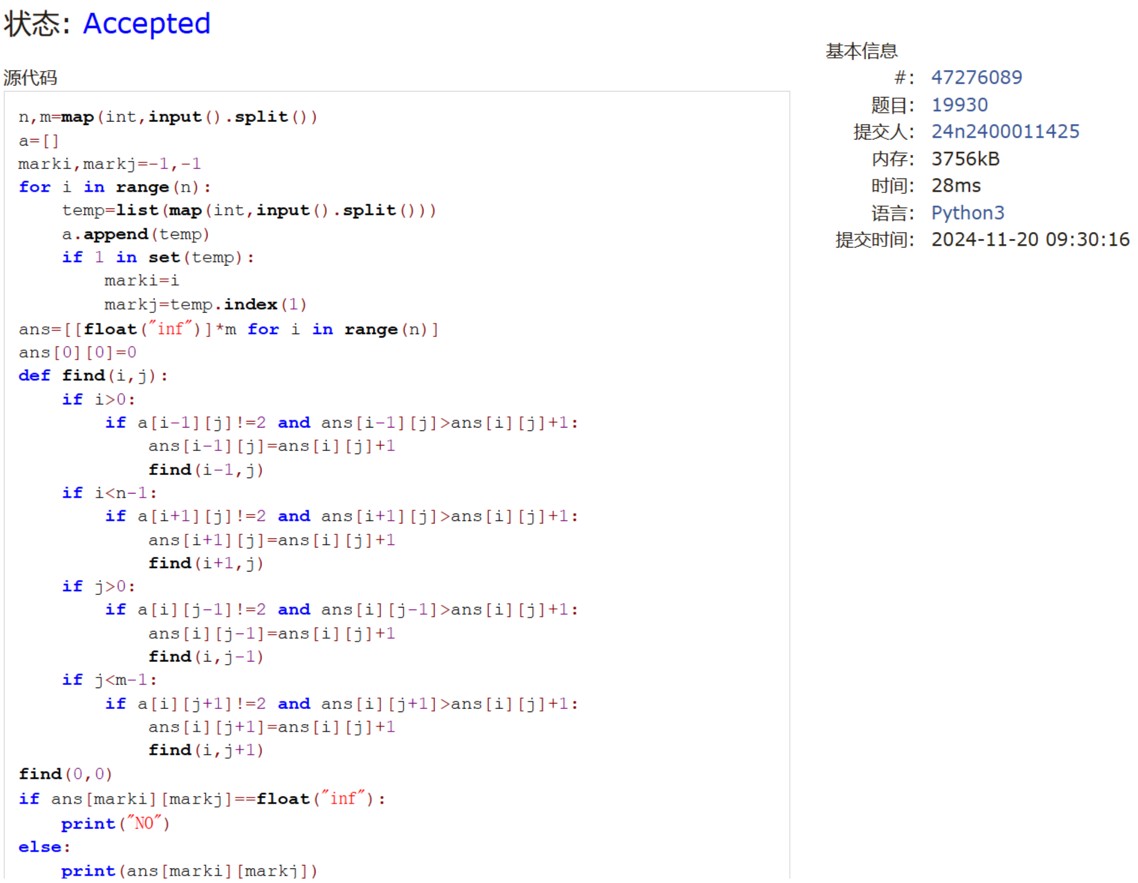
思路：

深搜，如果从i，j点去邻近的点比原本更近，那么就去搜那个点。

代码：

```
n,m=map(int,input().split())
a=[]
marki,markj=-1,-1
for i in range(n):
    temp=list(map(int,input().split()))
    a.append(temp)
    if 1 in set(temp):
        marki=i
        markj=temp.index(1)
ans=[[float("inf")]*m for i in range(n)]
ans[0][0]=0
def find(i,j):
    if i>0:
        if a[i-1][j]!=2 and ans[i-1][j]>ans[i][j]+1:
            ans[i-1][j]=ans[i][j]+1
            find(i-1,j)
    if i<n-1:
        if a[i+1][j]!=2 and ans[i+1][j]>ans[i][j]+1:
            ans[i+1][j]=ans[i][j]+1
            find(i+1,j)
    if j>0:
        if a[i][j-1]!=2 and ans[i][j-1]>ans[i][j]+1:
            ans[i][j-1]=ans[i][j]+1
            find(i,j-1)
    if j<m-1:
        if a[i][j+1]!=2 and ans[i][j+1]>ans[i][j]+1:
            ans[i][j+1]=ans[i][j]+1
            find(i,j+1)
find(0,0)
if ans[marki][markj]==float("inf"):
    print("NO")
else:
    print(ans[marki][markj])
```

代码运行截图 == （至少包含有"Accepted"） ==



状态: Accepted

源代码

```
n,m=map(int,input().split())
a=[]
marki,markj=-1,-1
for i in range(n):
    temp=list(map(int,input().split()))
    a.append(temp)
    if 1 in set(temp):
        marki=i
        markj=temp.index(1)
ans=[[float("inf")]*m for i in range(n)]
ans[0][0]=0
def find(i,j):
    if i>0:
        if a[i-1][j]!=2 and ans[i-1][j]>ans[i][j]+1:
            ans[i-1][j]=ans[i][j]+1
            find(i-1,j)
    if i<n-1:
        if a[i+1][j]!=2 and ans[i+1][j]>ans[i][j]+1:
            ans[i+1][j]=ans[i][j]+1
            find(i+1,j)
    if j>0:
        if a[i][j-1]!=2 and ans[i][j-1]>ans[i][j]+1:
            ans[i][j-1]=ans[i][j]+1
            find(i,j-1)
    if j<m-1:
        if a[i][j+1]!=2 and ans[i][j+1]>ans[i][j]+1:
            ans[i][j+1]=ans[i][j]+1
            find(i,j+1)
find(0,0)
if ans[marki][markj]==float("inf"):
    print("NO")
else:
    print(ans[marki][markj])
```

基本信息

#: 47276089
题目: 19930
提交人: 24n2400011425
内存: 3756kB
时间: 28ms
语言: Python3
提交时间: 2024-11-20 09:30:16

04123: 马走日

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

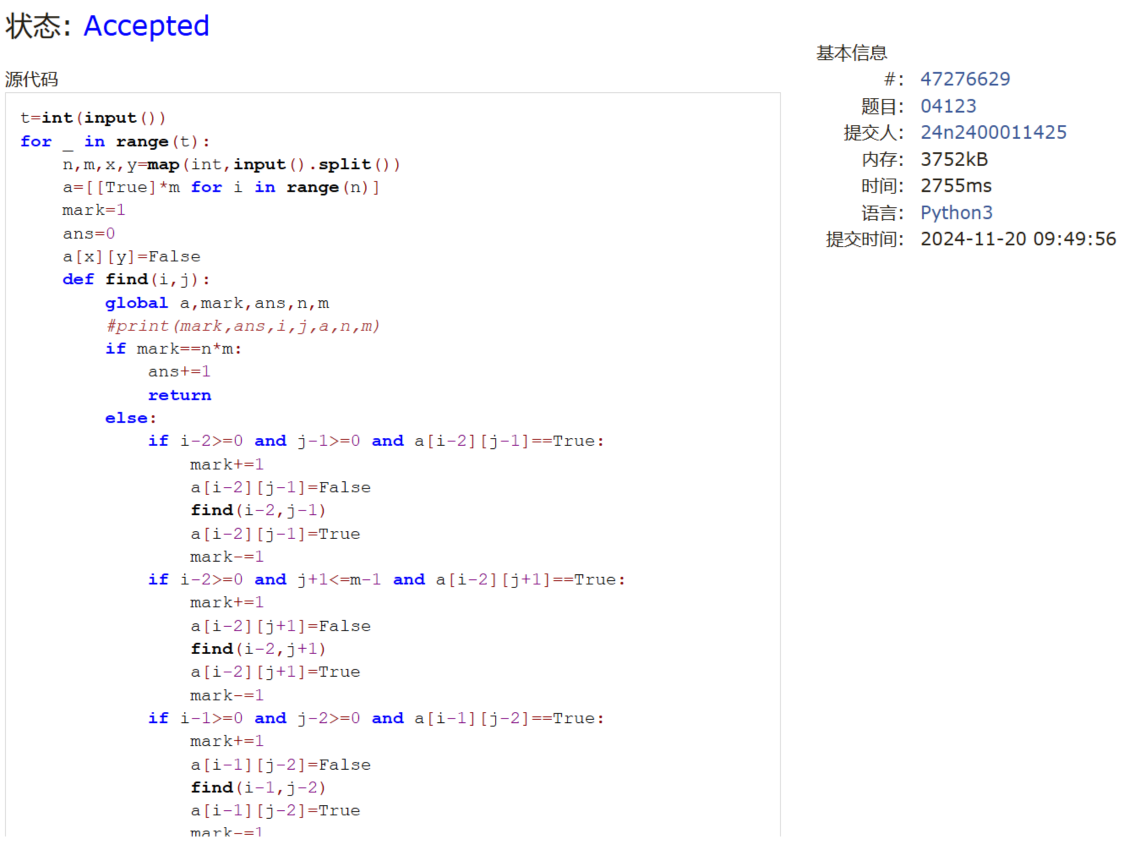
思路：

遍历每个可以去的格点，标记为否，遍历完了记得标记为真。利用一个mark来表示已经遍历了多少格点，当mark=nm时表示遍历完，ans+=1

代码：

```
t=int(input())
for _ in range(t):
    n,m,x,y=map(int,input().split())
    a=[[True]*m for i in range(n)]
    mark=1
    ans=0
    a[x][y]=False
    def find(i,j):
        global a,mark,ans,n,m
        #print(mark,ans,i,j,a,n,m)
        if mark==n*m:
            ans+=1
            return
        else:
            if i-2>=0 and j-1>=0 and a[i-2][j-1]==True:
                mark+=1
                a[i-2][j-1]=False
                find(i-2,j-1)
                a[i-2][j-1]=True
                mark-=1
            if i-2>=0 and j+1<=m-1 and a[i-2][j+1]==True:
                mark+=1
                a[i-2][j+1]=False
                find(i-2,j+1)
                a[i-2][j+1]=True
                mark-=1
            if i-1>=0 and j-2>=0 and a[i-1][j-2]==True:
                mark+=1
                a[i-1][j-2]=False
                find(i-1,j-2)
                a[i-1][j-2]=True
                mark-=1
            if i-1>=0 and j+2<=m-1 and a[i-1][j+2]==True:
                mark+=1
                a[i-1][j+2]=False
                find(i-1,j+2)
                a[i-1][j+2]=True
                mark-=1
            if i+2<=n-1 and j-1>=0 and a[i+2][j-1]==True:
                mark+=1
                a[i+2][j-1]=False
                find(i+2,j-1)
                a[i+2][j-1]=True
                mark-=1
            if i+2<=n-1 and j+1<=m-1 and a[i+2][j+1]==True:
                mark+=1
                a[i+2][j+1]=False
                find(i+2,j+1)
                a[i+2][j+1]=True
                mark-=1
            if i+1<=n-1 and j-2>=0 and a[i+1][j-2]==True:
                mark+=1
                a[i+1][j-2]=False
                find(i+1,j-2)
                a[i+1][j-2]=True
                mark-=1
            if i+1<=n-1 and j+2<=m-1 and a[i+1][j+2]==True:
                mark+=1
                a[i+1][j+2]=False
                find(i+1,j+2)
                a[i+1][j+2]=True
                mark-=1
    find(x,y)
    print(ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
t=int(input())
for _ in range(t):
    n,m,x,y=map(int,input().split())
    a=[[True]*m for i in range(n)]
    mark=1
    ans=0
    a[x][y]=False
    def find(i,j):
        global a,mark,ans,n,m
        #print(mark,ans,i,j,a,n,m)
        if mark==n*m:
            ans+=1
            return
        else:
            if i-2>=0 and j-1>=0 and a[i-2][j-1]==True:
                mark+=1
                a[i-2][j-1]=False
                find(i-2,j-1)
                a[i-2][j-1]=True
                mark-=1
            if i-2>=0 and j+1<=m-1 and a[i-2][j+1]==True:
                mark+=1
                a[i-2][j+1]=False
                find(i-2,j+1)
                a[i-2][j+1]=True
                mark-=1
            if i-1>=0 and j-2>=0 and a[i-1][j-2]==True:
                mark+=1
                a[i-1][j-2]=False
                find(i-1,j-2)
                a[i-1][j-2]=True
                mark-=1
```

基本信息

#:	47276629
题目:	04123
提交人:	24n2400011425
内存:	3752kB
时间:	2755ms
语言:	Python3
提交时间:	2024-11-20 09:49:56

sy316: 矩阵最大权值路径

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

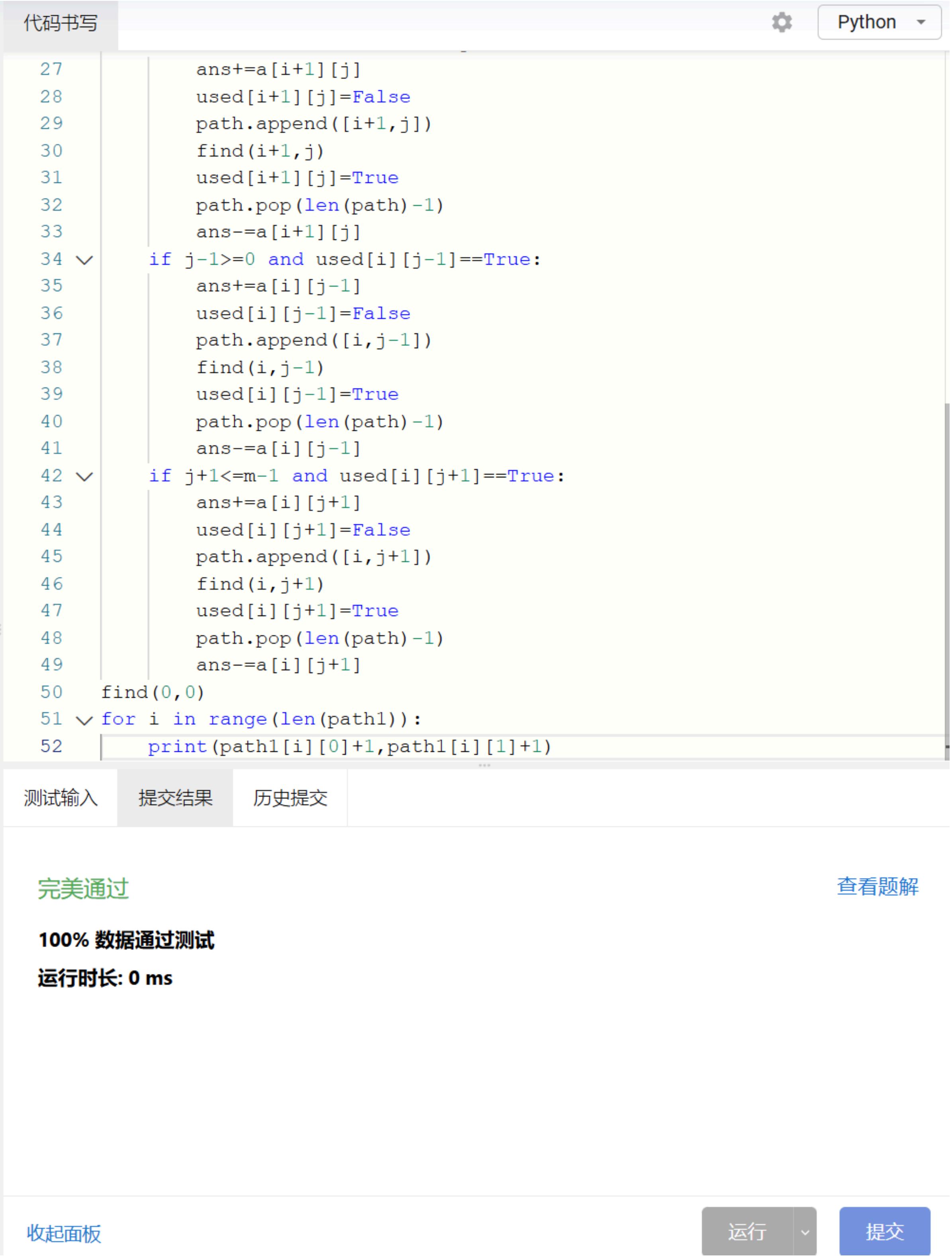
思路:

遍历所有可能路径，找出此时到右下角数字权重和，如果大于之前的数则记录此时的路径

代码:

```python
n,m=map(int,input().split())
a=[]
for _ in range(n):
    a.append(list(map(int,input().split())))
used=[[True]*m for i in range(n)]
used[0][0]=False
path=[[0,0]]
ans=a[0][0]
mark=float("-inf")
path1=[]
def find(i,j):
    global ans,mark,path1
    if i==n-1 and j==m-1:
        if ans>mark:
            path1=[path[i] for i in range(len(path))]
            mark=ans
        return
    if i-1>=0 and used[i-1][j]==True:
        ans+=a[i-1][j]
        used[i-1][j]=False
        path.append([i-1,j])
        find(i-1,j)
        used[i-1][j]=True
        ans-=a[i-1][j]
        path.pop(len(path)-1)
    if i+1<=n-1 and used[i+1][j]==True:
        ans+=a[i+1][j]
        used[i+1][j]=False
        path.append([i+1,j])
        find(i+1,j)
        used[i+1][j]=True
        path.pop(len(path)-1)
        ans-=a[i+1][j]
    if j-1>=0 and used[i][j-1]==True:
        ans+=a[i][j-1]
        used[i][j-1]=False
        path.append([i,j-1])
        find(i,j-1)
        used[i][j-1]=True
        path.pop(len(path)-1)
        ans-=a[i][j-1]
    if j+1<=m-1 and used[i][j+1]==True:
        ans+=a[i][j+1]
        used[i][j+1]=False
        path.append([i,j+1])
        find(i,j+1)
        used[i][j+1]=True
        path.pop(len(path)-1)
        ans-=a[i][j+1]
find(0,0)
for i in range(len(path1)):
    print(path1[i][0]+1,path1[i][1]+1)
```

代码运行截图 （至少包含有"Accepted"）



```python
        ans+=a[i+1][j]
        used[i+1][j]=False
        path.append([i+1,j])
        find(i+1,j)
        used[i+1][j]=True
        path.pop(len(path)-1)
        ans-=a[i+1][j]
    if j-1>=0 and used[i][j-1]==True:
        ans+=a[i][j-1]
        used[i][j-1]=False
        path.append([i,j-1])
        find(i,j-1)
        used[i][j-1]=True
        path.pop(len(path)-1)
        ans-=a[i][j-1]
    if j+1<=m-1 and used[i][j+1]==True:
        ans+=a[i][j+1]
        used[i][j+1]=False
        path.append([i,j+1])
        find(i,j+1)
        used[i][j+1]=True
        path.pop(len(path)-1)
        ans-=a[i][j+1]
find(0,0)
for i in range(len(path1)):
    print(path1[i][0]+1,path1[i][1]+1)
```

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/

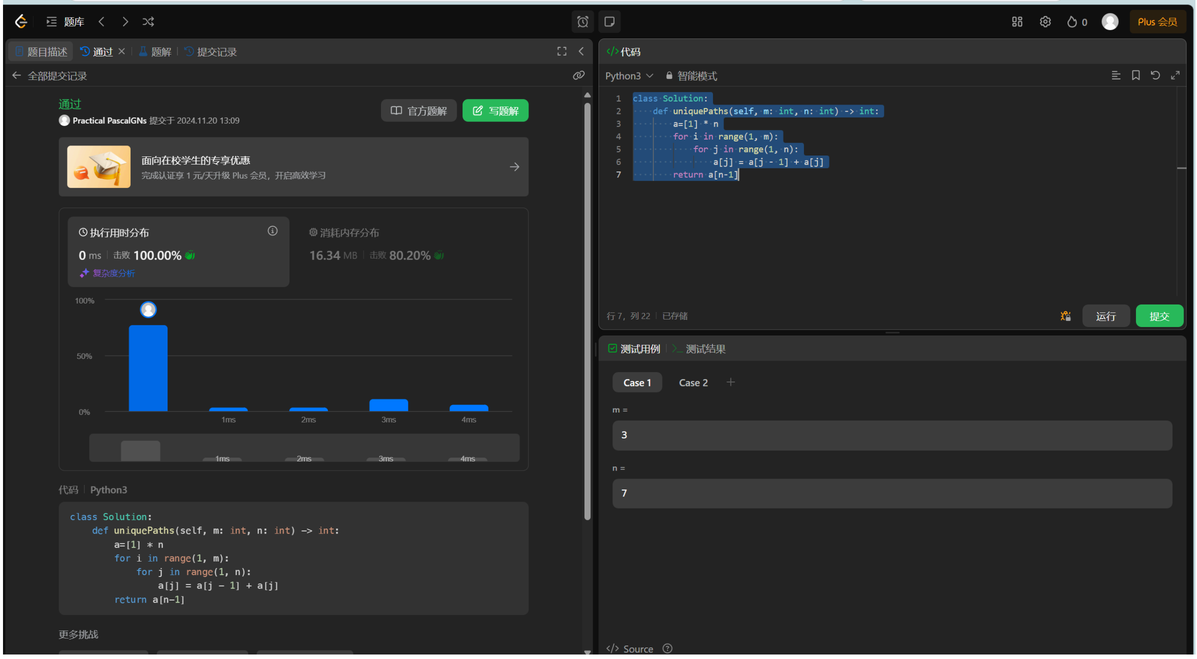
思路：

这个格子的最大路径等于上面和左边的格子的最小值+1，这就是转移方程。发现我们其实只用了上一行的数据所以我们还可以减小一个储存维度。

代码：

```
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        a=[1] * n
        for i in range(1, m):
            for j in range(1, n):
                a[j] = a[j - 1] + a[j]
        return a[n-1]
```

代码运行截图 （至少包含有"Accepted"）



通过
Practical PascalGNs 提交于 2024.11.20 13:09

0 ms | 击败 100.00%
16.34 MB | 击败 80.20%

代码 | Python3

```
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        a=[1] * n
        for i in range(1, m):
            for j in range(1, n):
                a[j] = a[j - 1] + a[j]
        return a[n-1]
```

Python3 智能模式

```
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        a=[1] * n
        for i in range(1, m):
            for j in range(1, n):
                a[j] = a[j - 1] + a[j]
        return a[n-1]
```

行 7，列 22 | 已存储

Case 1 Case 2

m =
3

n =
7

sy358: 受到祝福的平方

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

思路：

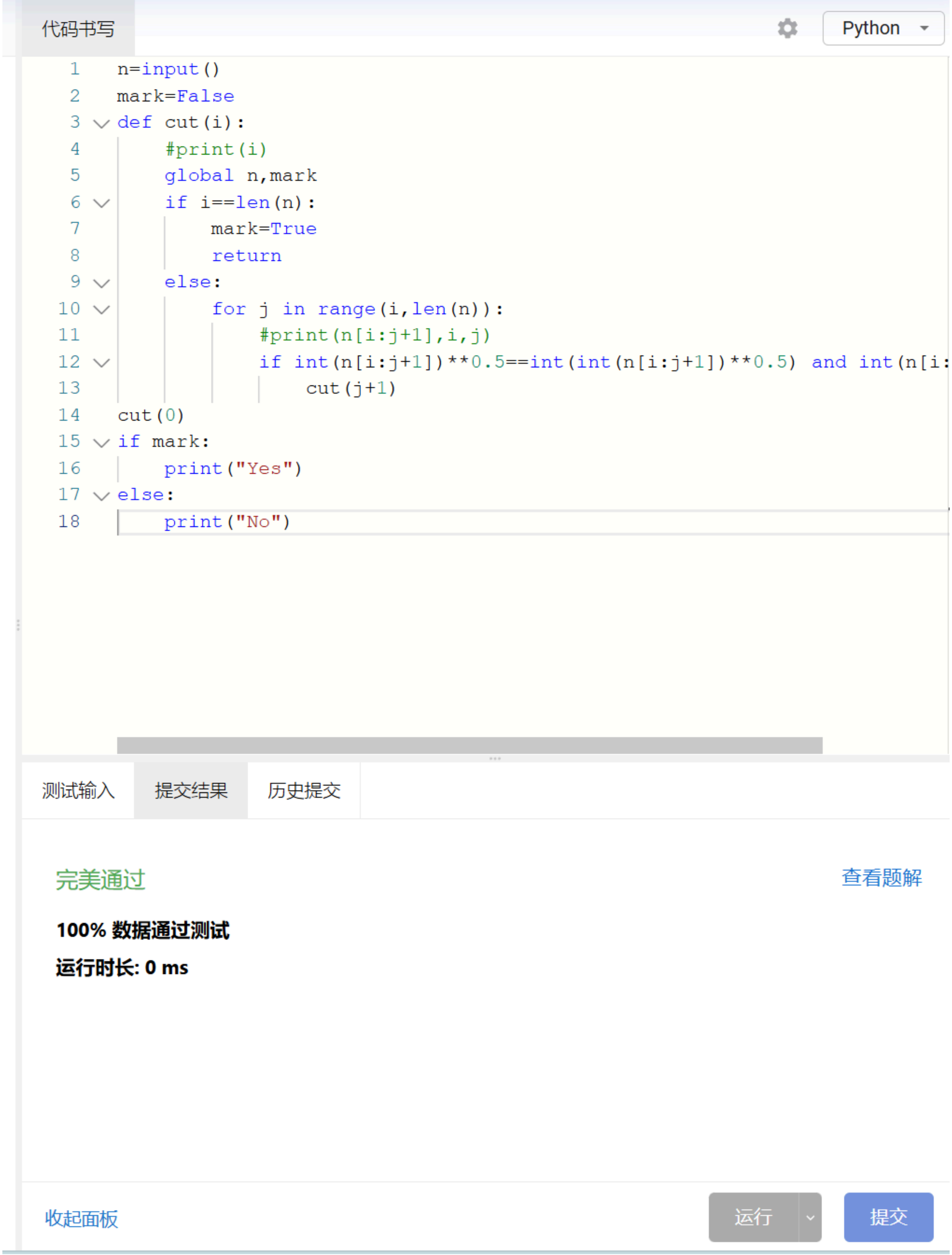
考虑上一个切割在i处的切割，则从i往后遍历，只要存在可以切割的方式那就让他切一下让后继续遍历。如果能把所有的东西都切完，也就是索引到n的末位数了，那就给一个标记表示可以分割。

代码：

```
n=input()
mark=False
def cut(i):
    #print(i)
    global n,mark
    if i==len(n):
        mark=True
        return
    else:
        for j in range(i,len(n)):
            #print(n[i:j+1],i,j)
            if int(n[i:j+1])**0.5==int(int(n[i:j+1])**0.5) and int(n[i:j+1])!=0:
                cut(j+1)
```

```python
cut(0)
if mark:
    print("Yes")
else:
    print("No")
```

代码运行截图 （至少包含有"Accepted"）



```python
n=input()
mark=False
def cut(i):
    #print(i)
    global n,mark
    if i==len(n):
        mark=True
        return
    else:
        for j in range(i,len(n)):
            #print(n[i:j+1],i,j)
            if int(n[i:j+1])**0.5==int(int(n[i:j+1])**0.5) and int(n[i:
                cut(j+1)
cut(0)
if mark:
    print("Yes")
else:
    print("No")
```

完美通过

查看题解

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。

dfs和bfs还是有很多东西可以挖掘的，感觉挺奇妙的。看到群里感觉大家都学了好多东西，有点惭愧，因为我只是学习了老师讲的知识和完成了老师布置上课的任务吗，感觉没多少时间和兴趣去拓展......但还是感谢这门课能让我看到一个不一样的计算机世界，即使我以后没去搞和他相关的东西我觉得还是挺有价值的。目前还是在跟进每日选做。