Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 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

18160: 最大连通域面积

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

思路：

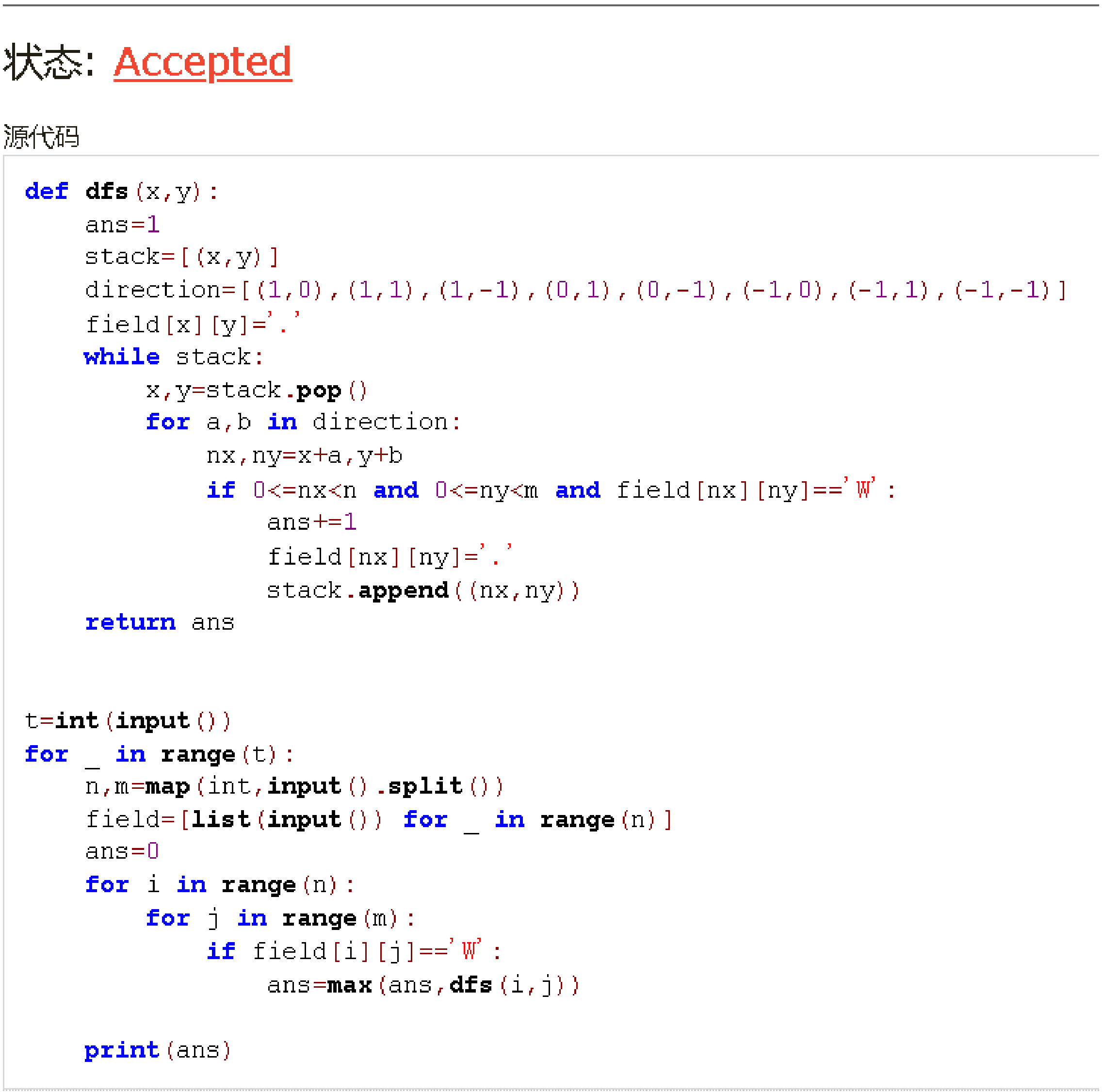
代码：

```
def dfs(x,y):
    ans=1
    stack=[(x,y)]
    direction=[(1,0),(1,1),(1,-1),(0,1),(0,-1),(-1,0),(-1,1),(-1,-1)]
    field[x][y]='.'
    while stack:
        x,y=stack.pop()
        for a,b in direction:
            nx,ny=x+a,y+b
            if 0<=nx<n and 0<=ny<m and field[nx][ny]=='W':
                ans+=1
                field[nx][ny]='.'
                stack.append((nx,ny))
    return ans


t=int(input())
for _ in range(t):
    n,m=map(int,input().split())
    field=[list(input()) for _ in range(n)]
    ans=0
    for i in range(n):
        for j in range(m):
            if field[i][j]=='W':
                ans=max(ans,dfs(i,j))

    print(ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
def dfs(x,y):
    ans=1
    stack=[(x,y)]
    direction=[(1,0),(1,1),(1,-1),(0,1),(0,-1),(-1,0),(-1,1),(-1,-1)]
    field[x][y]='.'
    while stack:
        x,y=stack.pop()
        for a,b in direction:
            nx,ny=x+a,y+b
            if 0<=nx<n and 0<=ny<m and field[nx][ny]=='W':
                ans+=1
                field[nx][ny]='.'
                stack.append((nx,ny))
    return ans


t=int(input())
for _ in range(t):
    n,m=map(int,input().split())
    field=[list(input()) for _ in range(n)]
    ans=0
    for i in range(n):
        for j in range(m):
            if field[i][j]=='W':
                ans=max(ans,dfs(i,j))

    print(ans)
```

19930: 寻宝

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

思路：虽然是bfs，但还是用dfs写的，而且没加#pylint:skip-file前又compile error了

代码：

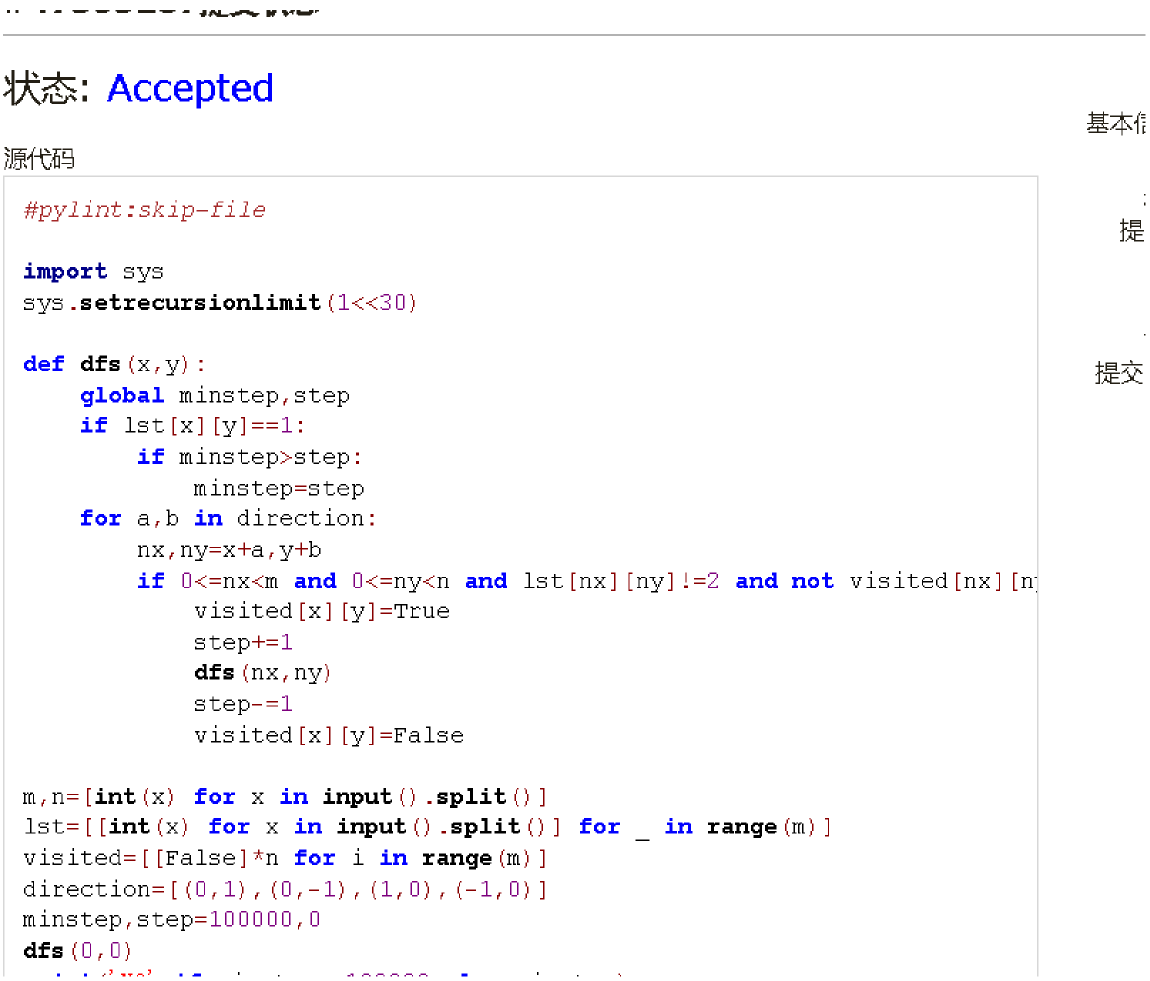
```
#pylint:skip-file

import sys
sys.setrecursionlimit(1<<30)
```

```
def dfs(x,y):
    global minstep,step
    if lst[x][y]==1:
        if minstep>step:
            minstep=step
    for a,b in direction:
        nx,ny=x+a,y+b
        if 0<=nx<m and 0<=ny<n and lst[nx][ny]!=2 and not visited[nx][ny]:
            visited[x][y]=True
            step+=1
            dfs(nx,ny)
            step-=1
            visited[x][y]=False

m,n=[int(x) for x in input().split()]
lst=[[int(x) for x in input().split()] for _ in range(m)]
visited=[[False]*n for i in range(m)]
direction=[(0,1),(0,-1),(1,0),(-1,0)]
minstep,step=100000,0
dfs(0,0)
print('NO' if minstep==100000 else minstep)
```

代码运行截图 == （至少包含有"Accepted"） ==



状态: Accepted

源代码

```
#pylint:skip-file

import sys
sys.setrecursionlimit(1<<30)

def dfs(x,y):
    global minstep,step
    if lst[x][y]==1:
        if minstep>step:
            minstep=step
    for a,b in direction:
        nx,ny=x+a,y+b
        if 0<=nx<m and 0<=ny<n and lst[nx][ny]!=2 and not visited[nx][n
            visited[x][y]=True
            step+=1
            dfs(nx,ny)
            step-=1
            visited[x][y]=False

m,n=[int(x) for x in input().split()]
lst=[[int(x) for x in input().split()] for _ in range(m)]
visited=[[False]*n for i in range(m)]
direction=[(0,1),(0,-1),(1,0),(-1,0)]
minstep,step=100000,0
dfs(0,0)
```

04123: 马走日

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

思路：为什么加个#pylint:skip-file 就不会出现compile error了，思考ing

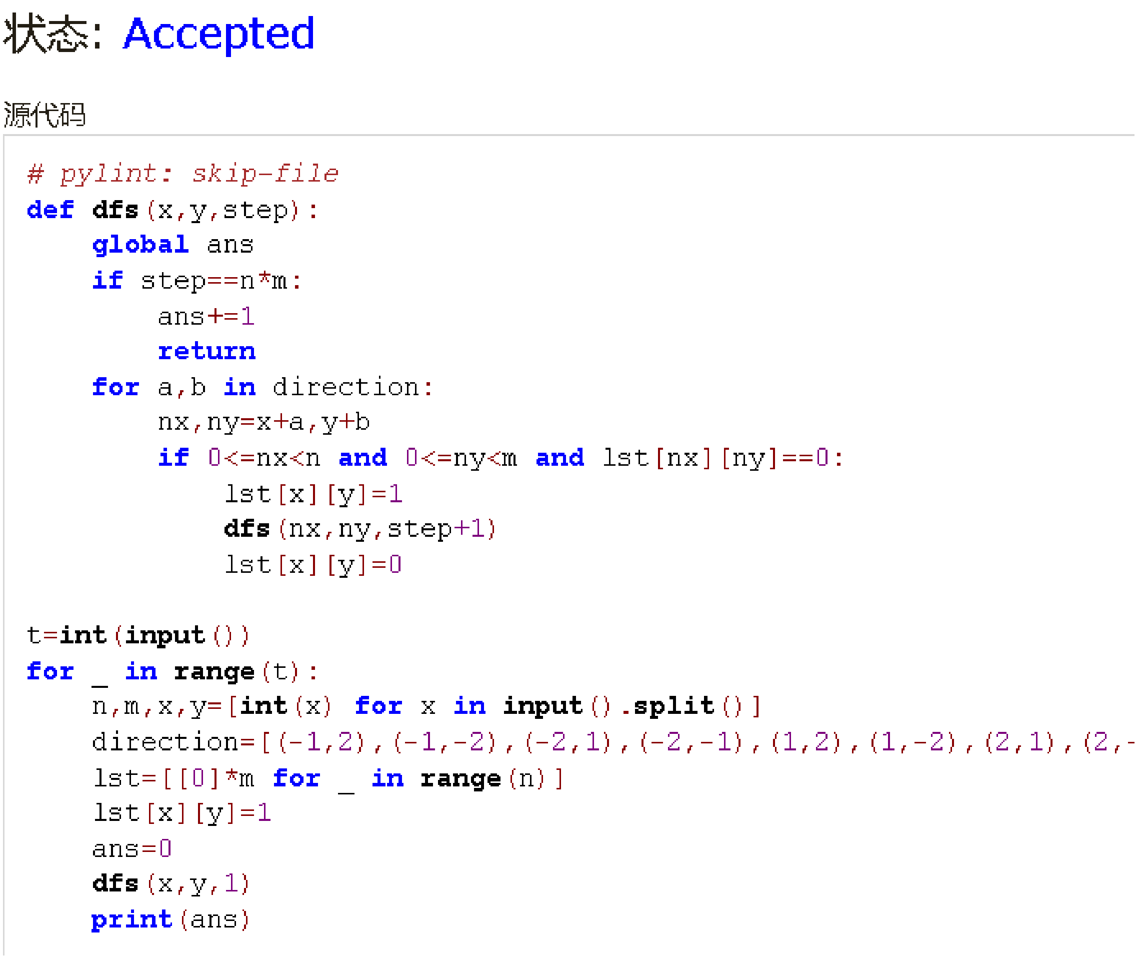
ai解释没看懂......

代码：

```python
# pylint: skip-file
def dfs(x,y,step):
    global ans
    if step==n*m:
        ans+=1
        return
    for a,b in direction:
        nx,ny=x+a,y+b
        if 0<=nx<n and 0<=ny<m and lst[nx][ny]==0:
            lst[x][y]=1
            dfs(nx,ny,step+1)
            lst[x][y]=0

t=int(input())
for _ in range(t):
    n,m,x,y=[int(x) for x in input().split()]
    direction=[(-1,2),(-1,-2),(-2,1),(-2,-1),(1,2),(1,-2),(2,1),(2,-1)]
    lst=[[0]*m for _ in range(n)]
    lst[x][y]=1
    ans=0
    dfs(x,y,1)
    print(ans)
```

代码运行截图（至少包含有"Accepted"）



状态: Accepted

源代码

```
# pylint: skip-file
def dfs(x,y,step):
    global ans
    if step==n*m:
        ans+=1
        return
    for a,b in direction:
        nx,ny=x+a,y+b
        if 0<=nx<n and 0<=ny<m and lst[nx][ny]==0:
            lst[x][y]=1
            dfs(nx,ny,step+1)
            lst[x][y]=0

t=int(input())
for _ in range(t):
    n,m,x,y=[int(x) for x in input().split()]
    direction=[(-1,2),(-1,-2),(-2,1),(-2,-1),(1,2),(1,-2),(2,1),(2,-
    lst=[[0]*m for _ in range(n)]
    lst[x][y]=1
    ans=0
    dfs(x,y,1)
    print(ans)
```

sy316: 矩阵最大权值路径

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

思路:

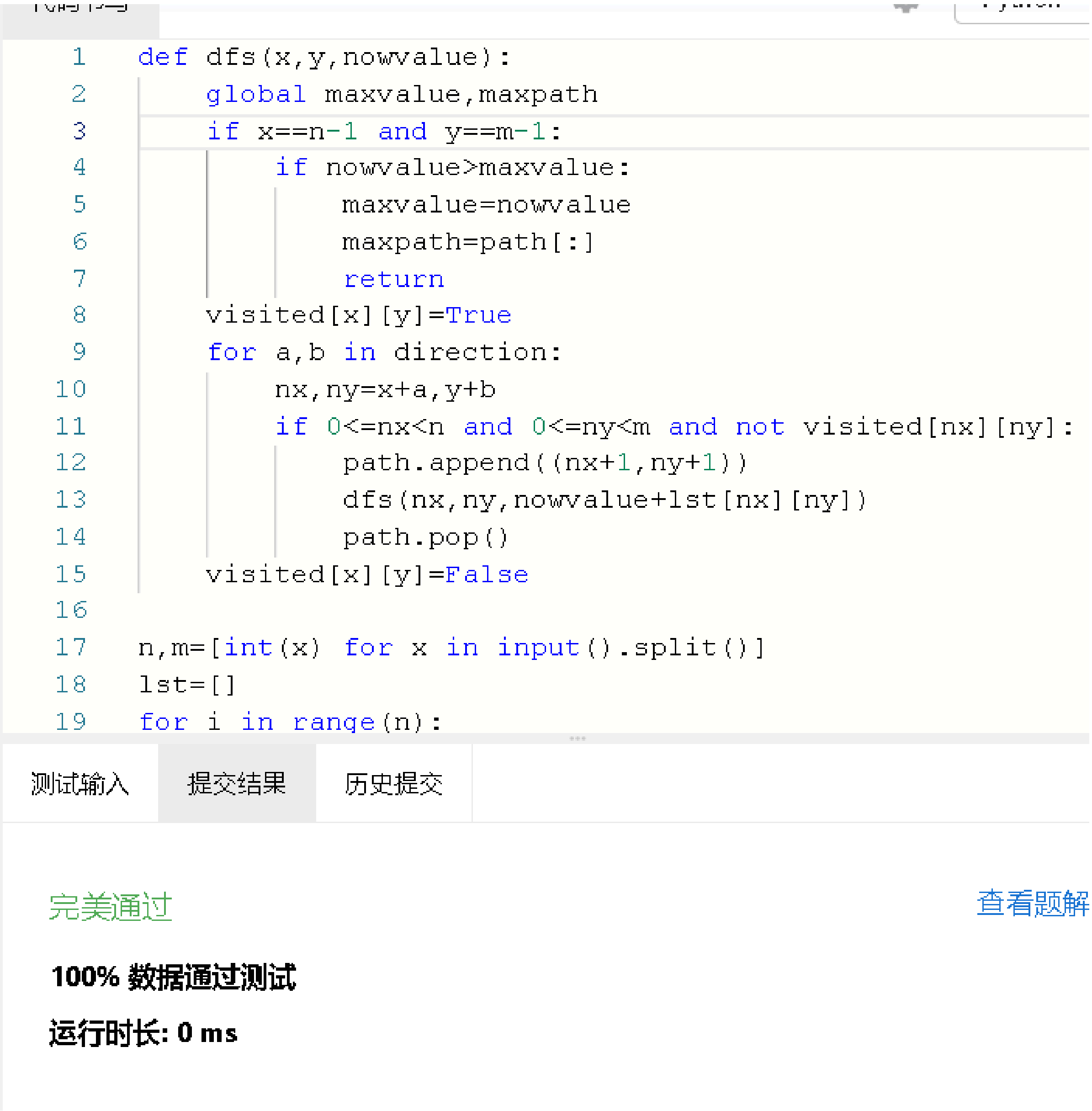
代码:

```
def dfs(x,y,nowvalue):
    global maxvalue,maxpath
    if x==n-1 and y==m-1:
        if nowvalue>maxvalue:
            maxvalue=nowvalue
            maxpath=path[:]
            return
    visited[x][y]=True
    for a,b in direction:
        nx,ny=x+a,y+b
        if 0<=nx<n and 0<=ny<m and not visited[nx][ny]:
            path.append((nx+1,ny+1))
            dfs(nx,ny,nowvalue+lst[nx][ny])
            path.pop()
    visited[x][y]=False

n,m=[int(x) for x in input().split()]
lst=[]
```

```
for i in range(n):
    lst.append([int(x) for x in input().split()])
maxvalue=float('-inf')
direction=[(0,1),(0,-1),(1,0),(-1,0)]
visited=[[False]*m for i in range(n)]
path=[(1,1)]
dfs(0,0,lst[0][0])
for i in maxpath:
    print(*i)
```

代码运行截图 （至少包含有"Accepted"）



```
def dfs(x,y,nowvalue):
    global maxvalue,maxpath
    if x==n-1 and y==m-1:
        if nowvalue>maxvalue:
            maxvalue=nowvalue
            maxpath=path[:]
            return
    visited[x][y]=True
    for a,b in direction:
        nx,ny=x+a,y+b
        if 0<=nx<n and 0<=ny<m and not visited[nx][ny]:
            path.append((nx+1,ny+1))
            dfs(nx,ny,nowvalue+lst[nx][ny])
            path.pop()
    visited[x][y]=False

n,m=[int(x) for x in input().split()]
lst=[]
for i in range(n):
```

完美通过

100% 数据通过测试

运行时长: 0 ms

LeetCode62.不同路径

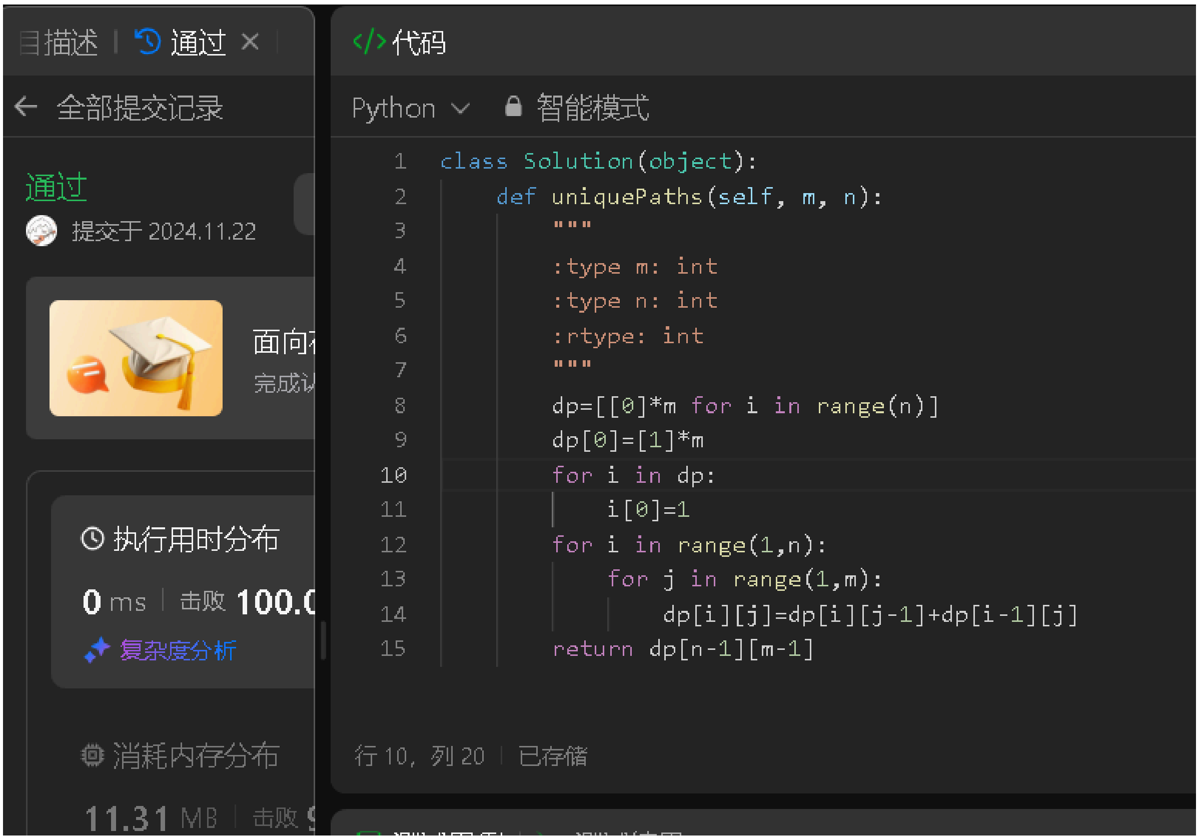
dp, https://leetcode.cn/problems/unique-paths/

思路：

代码：

```
class Solution(object):
    def uniquePaths(self, m, n):
        """
        :type m: int
        :type n: int
        :rtype: int
        """
        dp=[[0]*m for i in range(n)]
        dp[0]=[1]*m
        for i in dp:
            i[0]=1
        for i in range(1,n):
            for j in range(1,m):
                dp[i][j]=dp[i][j-1]+dp[i-1][j]
        return dp[n-1][m-1]
```

代码运行截图 （至少包含有"Accepted"）



通过

提交于 2024.11.22

面向
完成

执行用时分布

0 ms | 击败 100.0

消耗内存分布

11.31 MB | 击败 9

```
class Solution(object):
    def uniquePaths(self, m, n):
        """
        :type m: int
        :type n: int
        :rtype: int
        """
        dp=[[0]*m for i in range(n)]
        dp[0]=[1]*m
        for i in dp:
            i[0]=1
        for i in range(1,n):
            for j in range(1,m):
                dp[i][j]=dp[i][j-1]+dp[i-1][j]
        return dp[n-1][m-1]
```

行 10，列 20 | 已存储

sy358: 受到祝福的平方

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

思路：一开始忽略了必须是正整数的平方，导致100000没通过

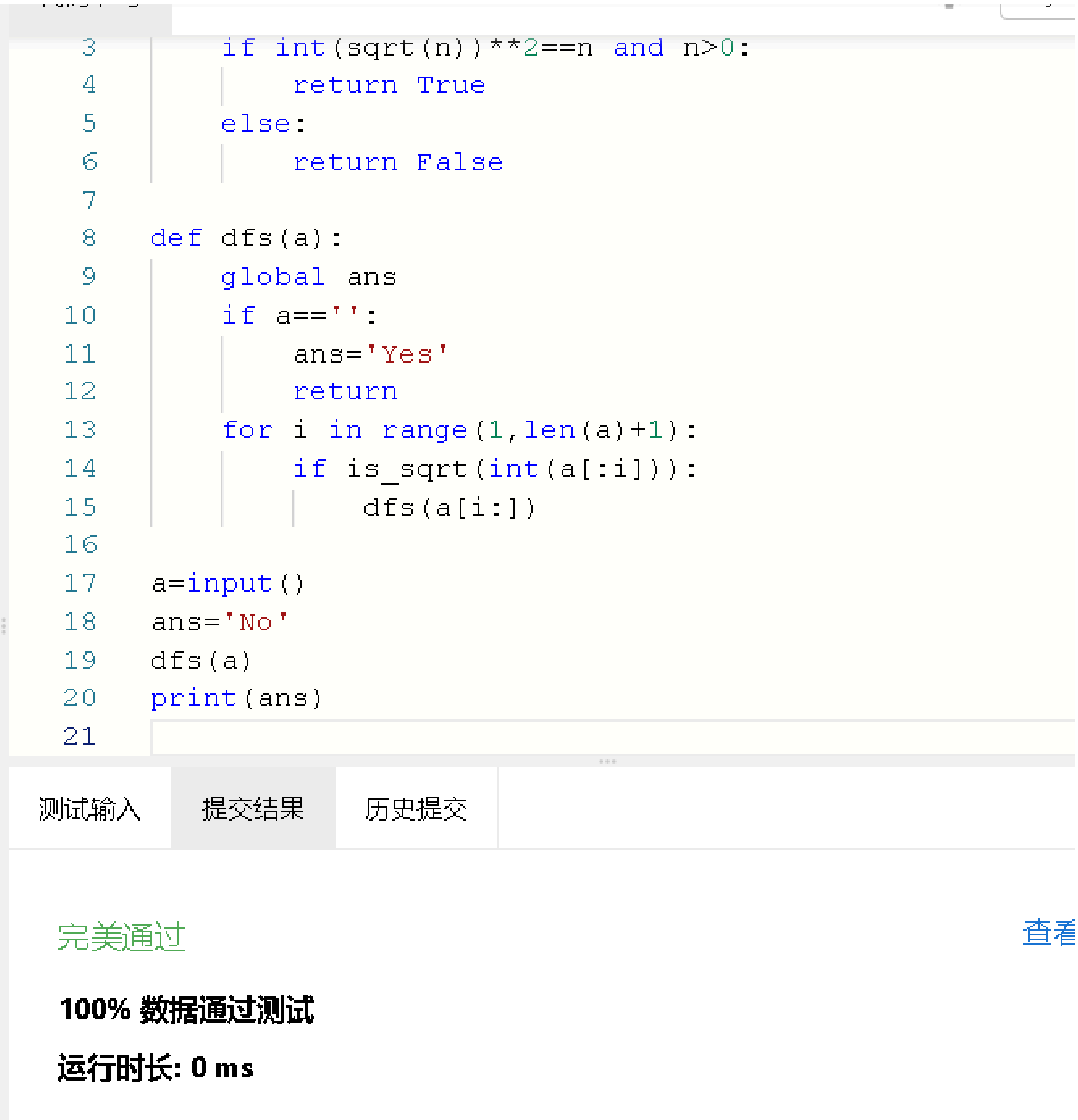
代码：

```python
from math import sqrt
def is_sqrt(n):
    if int(sqrt(n))**2==n and n>0:
        return True
    else:
        return False

def dfs(a):
    global ans
    if a=='':
        ans='Yes'
        return
    for i in range(1,len(a)+1):
        if is_sqrt(int(a[:i])):
            dfs(a[i:])

a=input()
ans='No'
dfs(a)
print(ans)
```

代码运行截图 （至少包含有"Accepted"）



```
    if int(sqrt(n))**2==n and n>0:
        return True
    else:
        return False

def dfs(a):
    global ans
    if a=='':
        ans='Yes'
        return
    for i in range(1,len(a)+1):
        if is_sqrt(int(a[:i])):
            dfs(a[i:])

a=input()
ans='No'
dfs(a)
print(ans)
```

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。

dp刚刚看到背包，先把dfs的课件看完了，感觉dfs的套路性都比较强，题目套一套模板都大差不差了，但对于每条代码的理解好像又不是那么透彻，可能因为还没做多少题？