Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by 万馨雅 城环

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

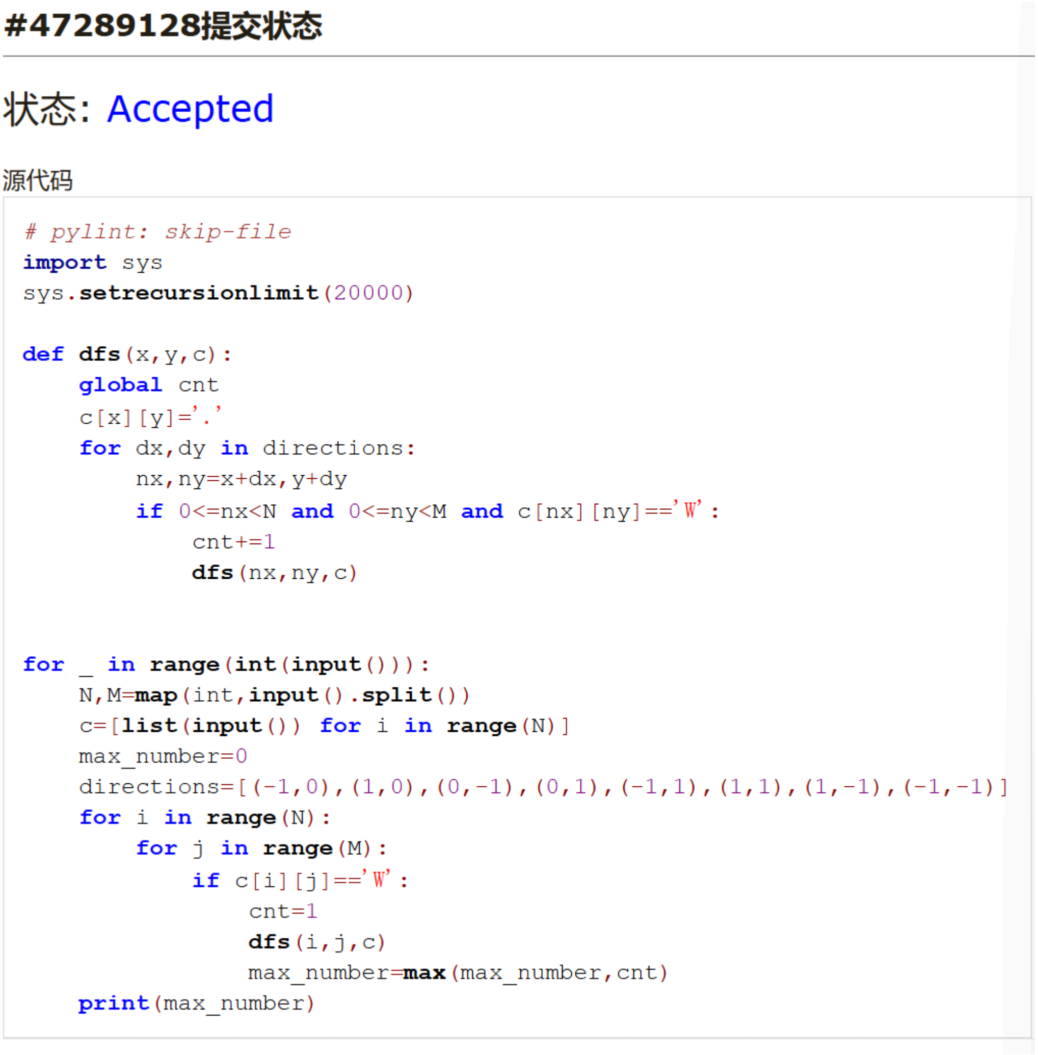
dfs similar, http://cs101.openjudge.cn/practice/18160

思路：

代码：

```python
import sys
sys.setrecursionlimit(20000)

def dfs(x,y,c):
    global cnt
    c[x][y]='.'
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if 0<=nx<N and 0<=ny<M and c[nx][ny]=='W':
            cnt+=1
            dfs(nx,ny,c)


for _ in range(int(input())):
    N,M=map(int,input().split())
    c=[list(input()) for i in range(N)]
    max_number=0
    directions=[(-1,0),(1,0),(0,-1),(0,1),(-1,1),(1,1),(1,-1),(-1,-1)]
    for i in range(N):
        for j in range(M):
            if c[i][j]=='W':
                cnt=1
                dfs(i,j,c)
                max_number=max(max_number,cnt)
    print(max_number)
```

代码运行截图 （至少包含有"Accepted"）



#47289128提交状态

状态: Accepted

源代码

```
# pylint: skip-file
import sys
sys.setrecursionlimit(20000)

def dfs(x,y,c):
    global cnt
    c[x][y]=’.’
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if 0<=nx<N and 0<=ny<M and c[nx][ny]==’W’:
            cnt+=1
            dfs(nx,ny,c)


for _ in range(int(input())):
    N,M=map(int,input().split())
    c=[list(input()) for i in range(N)]
    max_number=0
    directions=[(-1,0),(1,0),(0,-1),(0,1),(-1,1),(1,1),(1,-1),(-1,-1)]
    for i in range(N):
        for j in range(M):
            if c[i][j]==’W’:
                cnt=1
                dfs(i,j,c)
                max_number=max(max_number,cnt)
    print(max_number)
```

19930: 寻宝

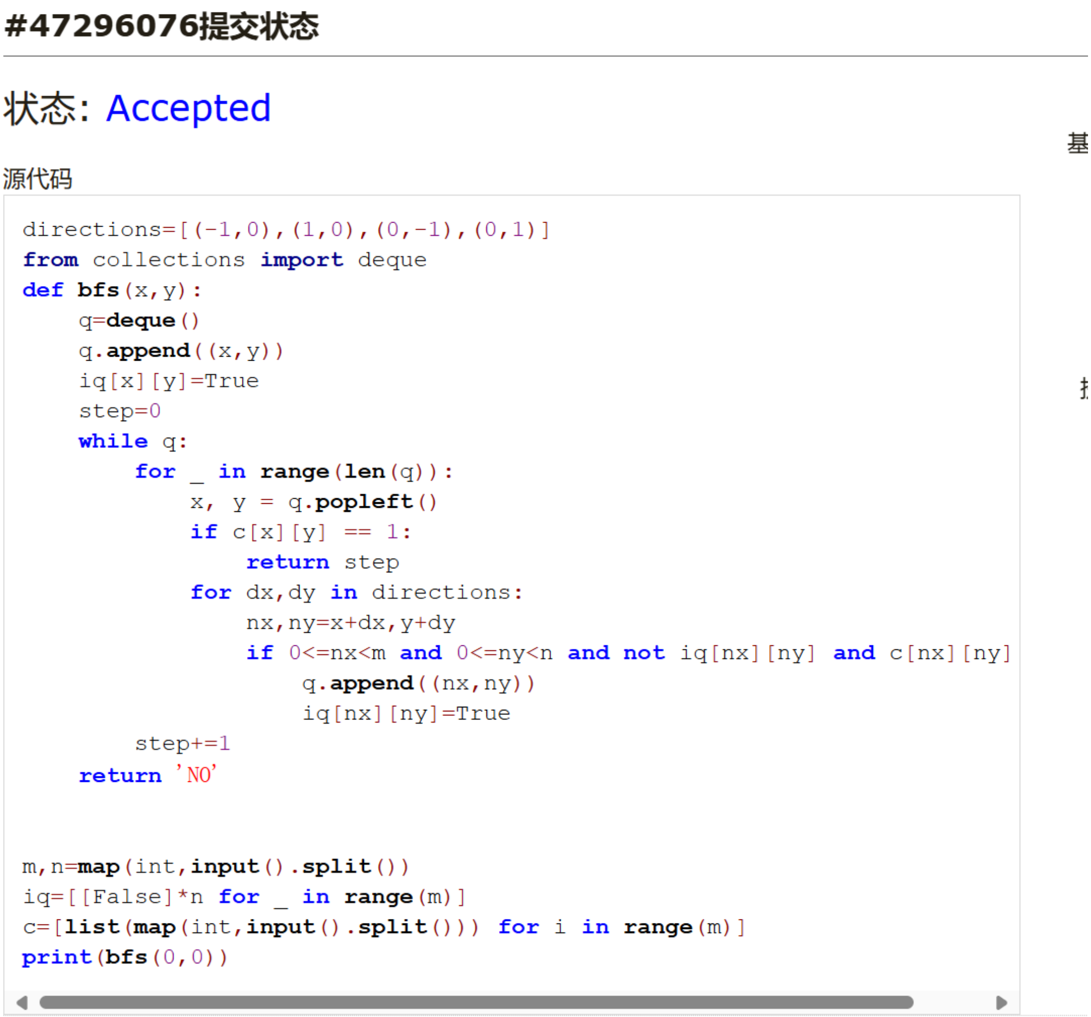
bfs, http://cs101.openjudge.cn/practice/19930

思路：

代码：

```
directions=[(-1,0),(1,0),(0,-1),(0,1)]
from collections import deque
def bfs(x,y):
    q=deque()
    q.append((x,y))
    iq[x][y]=True
    step=0
    while q:
        for _ in range(len(q)):
            x, y = q.popleft()
            if c[x][y] == 1:
                return step
            for dx,dy in directions:
                nx,ny=x+dx,y+dy
                if 0<=nx<m and 0<=ny<n and not iq[nx][ny] and c[nx][ny] !=2:
                    q.append((nx,ny))
                    iq[nx][ny]=True
        step+=1
    return 'NO'


m,n=map(int,input().split())
iq=[[False]*n for _ in range(m)]
c=[list(map(int,input().split())) for i in range(m)]
print(bfs(0,0))
```

代码运行截图 ==（至少包含有"Accepted"）==



#47296076提交状态

状态: Accepted

源代码

```
directions=[(-1,0),(1,0),(0,-1),(0,1)]
from collections import deque
def bfs(x,y):
    q=deque()
    q.append((x,y))
    iq[x][y]=True
    step=0
    while q:
        for _ in range(len(q)):
            x, y = q.popleft()
            if c[x][y] == 1:
                return step
            for dx,dy in directions:
                nx,ny=x+dx,y+dy
                if 0<=nx<m and 0<=ny<n and not iq[nx][ny] and c[nx][ny]
                    q.append((nx,ny))
                    iq[nx][ny]=True
        step+=1
    return ’NO’


m,n=map(int,input().split())
iq=[[False]*n for _ in range(m)]
c=[list(map(int,input().split())) for i in range(m)]
print(bfs(0,0))
```

04123: 马走日

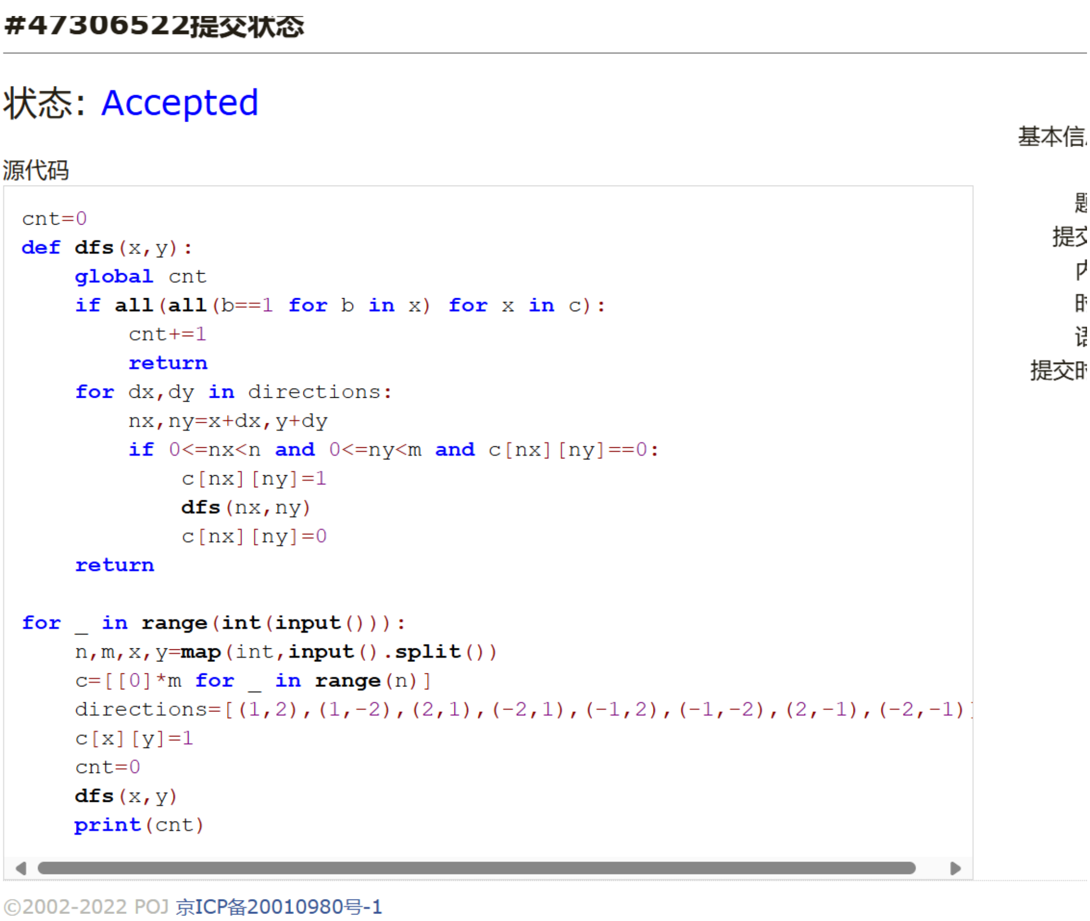
dfs, http://cs101.openjudge.cn/practice/04123

思路：

代码：

```
cnt=0
def dfs(x,y):
    global cnt
    if all(all(b==1 for b in x) for x in c):
        cnt+=1
        return
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if 0<=nx<n and 0<=ny<m and c[nx][ny]==0:
            c[nx][ny]=1
            dfs(nx,ny)
            c[nx][ny]=0
    return

for _ in range(int(input())):
    n,m,x,y=map(int,input().split())
    c=[[0]*m for _ in range(n)]
    directions=[(1,2),(1,-2),(2,1),(-2,1),(-1,2),(-1,-2),(2,-1),(-2,-1)]
    c[x][y]=1
    cnt=0
    dfs(x,y)
    print(cnt)
```

代码运行截图 （至少包含有"Accepted"）



#47306522提交状态

状态: Accepted

源代码

```
cnt=0
def dfs(x,y):
    global cnt
    if all(all(b==1 for b in x) for x in c):
        cnt+=1
        return
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if 0<=nx<n and 0<=ny<m and c[nx][ny]==0:
            c[nx][ny]=1
            dfs(nx,ny)
            c[nx][ny]=0
    return

for _ in range(int(input())):
    n,m,x,y=map(int,input().split())
    c=[[0]*m for _ in range(n)]
    directions=[(1,2),(1,-2),(2,1),(-2,1),(-1,2),(-1,-2),(2,-1),(-2,-1)
    c[x][y]=1
    cnt=0
    dfs(x,y)
    print(cnt)
```

©2002-2022 POJ 京ICP备20010980号-1

sy316: 矩阵最大权值路径

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

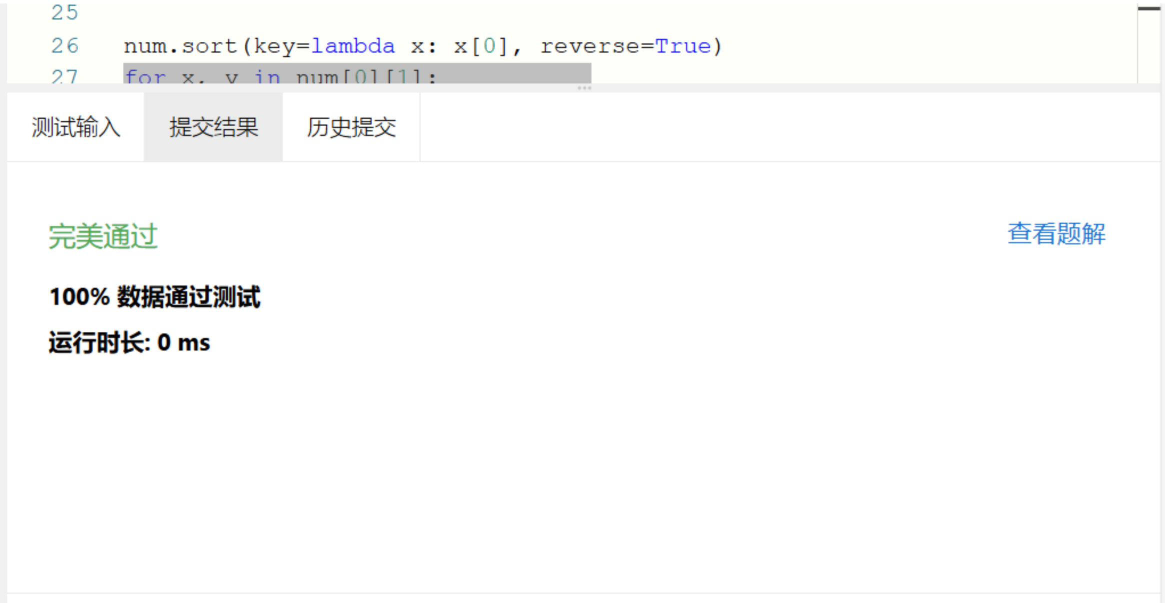
思路:

代码:

```
def dfs(cnt, x, y):
    global e
    global num
    if x == n and y == m:
        num.append((cnt, e.copy()))
        return
    for dx, dy in directions:
        nx, ny = x + dx, y + dy
        if 0 <= nx < n + 2 and 0 <= ny < m + 2 and not d[nx][ny] and c[nx][ny] != float('-inf'):
            d[nx][ny] = True
            e.append((nx, ny))
            dfs(cnt + c[nx][ny], nx, ny)
            d[nx][ny] = False
            e.pop()

n, m = map(int, input().split())
directions = [(1, 0), (-1, 0), (0, 1), (0, -1)]
```

```
c = [[float('-inf')] * (m + 2)] + [[float('-inf')] + list(map(int,
input().split())) + [float('-inf')] for _ in range(n)] + [[float('-inf')] * (m +
2)]
d = [[False] * (m + 2) for _ in range(n + 2)]
num = []
e = [(1, 1)]

d[1][1] = True
dfs(c[1][1], 1, 1)

num.sort(key=lambda x: x[0], reverse=True)
for x, y in num[0][1]:
    print(x, y)
```

代码运行截图 （至少包含有"Accepted"）



```
num.sort(key=lambda x: x[0], reverse=True)
for x, y in num[0][1]:
```

完美通过

100% 数据通过测试

运行时长: 0 ms

LeetCode62.不同路径

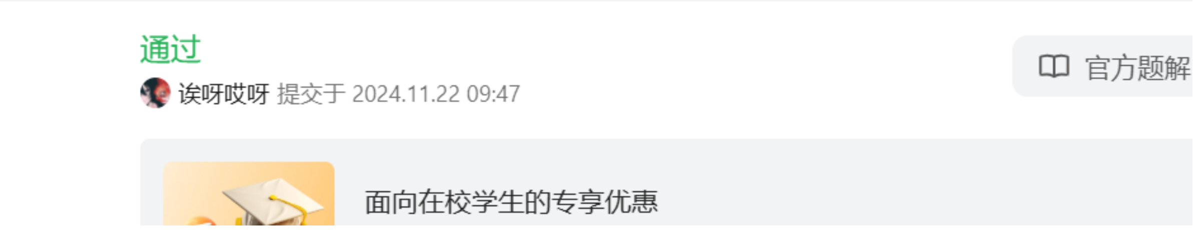
dp, https://leetcode.cn/problems/unique-paths/

思路：

代码：

```
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        dp = [[0] * n for _ in range(m)]

        for i in range(m):
            dp[i][0] = 1
        for j in range(n):
            dp[0][j] = 1


        for i in range(1, m):
            for j in range(1, n):
                dp[i][j] = dp[i-1][j] + dp[i][j-1]

        return(dp[-1][-1])
```

代码运行截图 （至少包含有"Accepted"）



通过

诶呀哎呀 提交于 2024.11.22 09:47

sy358: 受到祝福的平方

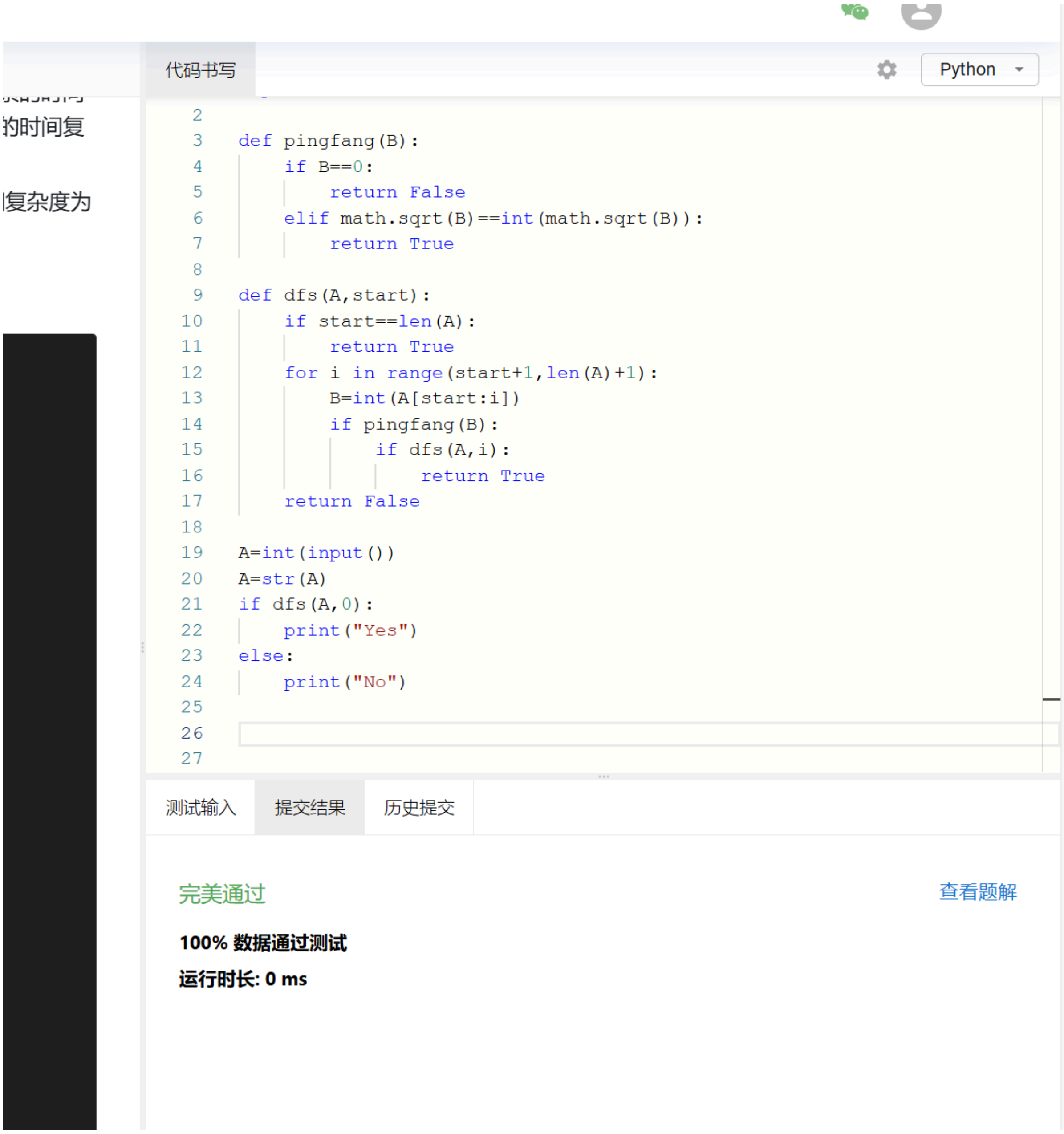
dfs, dp, https://sunnywhy.com/sfbj/8/3/539

思路：

代码：

```
import math

def pingfang(B):
    if B==0:
        return False
    elif math.sqrt(B)==int(math.sqrt(B)):
        return True

def dfs(A,start):
    if start==len(A):
        return True
    for i in range(start+1,len(A)+1):
        B=int(A[start:i])
        if pingfang(B):
            if dfs(A,i):
                return True
    return False

A=int(input())
A=str(A)
if dfs(A,0):
    print("Yes")
else:
    print("No")
```

代码运行截图 （至少包含有"Accepted"）



```python

def pingfang(B):
    if B==0:
        return False
    elif math.sqrt(B)==int(math.sqrt(B)):
        return True

def dfs(A,start):
    if start==len(A):
        return True
    for i in range(start+1,len(A)+1):
        B=int(A[start:i])
        if pingfang(B):
            if dfs(A,i):
                return True
    return False

A=int(input())
A=str(A)
if dfs(A,0):
    print("Yes")
else:
    print("No")


```

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

写起来没上周那么崩溃了，完全是模板的题目有点会了，但是稍微有点变化的又想不到了。受到祝福的平方在参考答案的思路后自己又写了一下，结果因为在验证是否平方数的时候把0包含进去了，导致一直没过。要好好看题啊！