Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by 陈冠宇 工学院

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：

代码：

```
def dfs(chizu, visited, x, y, N, M):
    directions = [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, -1), (1, 0),
(1, 1)]
    stack = [(x, y)]
    area = 0

    while stack:
        x, y = stack.pop()
        if not visited[x][y]:
            visited[x][y] = True
            area += 1
            for dx, dy in directions:
                nx, ny = x + dx, y + dy
                if 0 <= nx < N and 0 <= ny < M and chizu[nx][ny] == 1 and not
visited[nx][ny]:
                    stack.append((nx, ny))

    return area


def find_max_connected_area(N, M, chizu):
    visited = [[False for _ in range(M)] for _ in range(N)]
    max_area = 0
```

```
    for i in range(N):
        for j in range(M):
            if chizu[i][j] == 1 and not visited[i][j]:
                area = dfs(chizu, visited, i, j, N, M)
                max_area = max(max_area, area)

    return max_area

x = int(input())
for _ in range(x):
    n, m = map(int, input().split())
    chizu = [[0 for _ in range(m)] for _ in range(n)]
    count = 0

    for i in range(n):
        w = input().strip()
        for j in range(m):
            if w[j] == 'W':
                chizu[i][j] = 1
                count += 1

    if count == 0:
        print(0)
    else:
        max_area = find_max_connected_area(n, m, chizu)
        print(max_area)
```

代码运行截图 （至少包含有"Accepted"）

19930: 寻宝

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

思路：第一次写bfs，问ai了好久

代码：

```
from collections import deque
m,n=map(int,input().split())
chizu=[]
bfs_chizu=[[float('inf')for _ in range(n+2)] for _ in range(m+2)]
for i in range(0,m):
    chizu.append(list(map(int,input().split())))
bfs_chizu[1][1]=0
if chizu[0][0]==1:
    print(0)
    exit(0)
queue=deque([(1,1)])
while queue:
    x,y = queue.popleft()
    for dx, dy in [(1,0),(0,1),(-1,0),(0,-1)]:
```

```
        nx, ny = x + dx, y + dy
        if 1 <= nx <=m and 1 <= ny <= n and chizu[nx-1][ny-1]!=2:
            if bfs_chizu[nx][ny]>bfs_chizu[x][y]+1:
                bfs_chizu[nx][ny]=bfs_chizu[x][y]+1
                queue.append((nx, ny))
                if chizu[nx-1][ny-1] == 1:
                    print(bfs_chizu[nx][ny])
                    exit(0)
print("NO")
```

代码运行截图 ==（至少包含有"Accepted"）==

#47362032提交状态

查看　提交　统计　提问

状态: Accepted

源代码

```
from collections import deque
m,n=map(int,input().split())
chizu=[]
bfs_chizu=[[float('inf')for _ in range(n+2)] for _ in range(m+2)]
for i in range(0,m):
    chizu.append(list(map(int,input().split())))
bfs_chizu[1][1]=0
if chizu[0][0]==1:
    print(0)
    exit(0)
queue=deque([(1,1)])
while queue:
    x,y = queue.popleft()
    for dx, dy in [(1,0),(0,1),(-1,0),(0,-1)]:
        nx, ny = x + dx, y + dy
        if 1 <= nx <=m and 1 <= ny <= n and chizu[nx-1][ny-1]!=2:
            if bfs_chizu[nx][ny]>bfs_chizu[x][y]+1:
                bfs_chizu[nx][ny]=bfs_chizu[x][y]+1
                queue.append((nx, ny))
                if chizu[nx-1][ny-1] == 1:
                    print(bfs_chizu[nx][ny])
                    exit(0)
print("NO")
```

基本信息

#: 47362032
题目: 19930
提交人: 陈冠宇(24n2400011004)
内存: 3708kB
时间: 35ms
语言: Python3
提交时间: 2024-11-24 11:06:04

English　帮助　关于

04123: 马走日

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

思路:

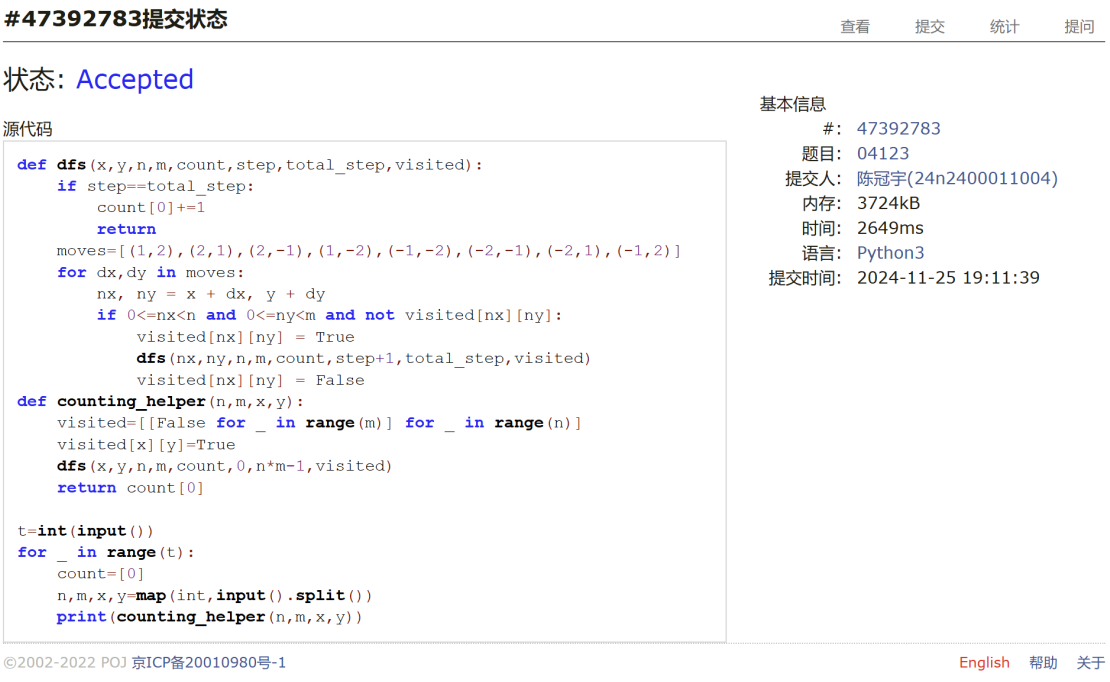
代码:

```
def dfs(x,y,n,m,count,step,total_step,visited):
    if step==total_step:
        count[0]+=1
        return
    moves=[(1,2),(2,1),(2,-1),(1,-2),(-1,-2),(-2,-1),(-2,1),(-1,2)]
    for dx,dy in moves:
        nx, ny = x + dx, y + dy
        if 0<=nx<n and 0<=ny<m and not visited[nx][ny]:
            visited[nx][ny] = True
            dfs(nx,ny,n,m,count,step+1,total_step,visited)
            visited[nx][ny] = False
```

```
def counting_helper(n,m,x,y):
    visited=[[False for _ in range(m)] for _ in range(n)]
    visited[x][y]=True
    dfs(x,y,n,m,count,0,n*m-1,visited)
    return count[0]

t=int(input())
for _ in range(t):
    count=[0]
    n,m,x,y=map(int,input().split())
    print(counting_helper(n,m,x,y))
```

代码运行截图 （至少包含有"Accepted"）



#47392783提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def dfs(x,y,n,m,count,step,total_step,visited):
    if step==total_step:
        count[0]+=1
        return
    moves=[(1,2),(2,1),(2,-1),(1,-2),(-1,-2),(-2,-1),(-2,1),(-1,2)]
    for dx,dy in moves:
        nx, ny = x + dx, y + dy
        if 0<=nx<n and 0<=ny<m and not visited[nx][ny]:
            visited[nx][ny] = True
            dfs(nx,ny,n,m,count,step+1,total_step,visited)
            visited[nx][ny] = False
def counting_helper(n,m,x,y):
    visited=[[False for _ in range(m)] for _ in range(n)]
    visited[x][y]=True
    dfs(x,y,n,m,count,0,n*m-1,visited)
    return count[0]

t=int(input())
for _ in range(t):
    count=[0]
    n,m,x,y=map(int,input().split())
    print(counting_helper(n,m,x,y))
```

基本信息

#: 47392783
题目: 04123
提交人: 陈冠宇(24n2400011004)
内存: 3724kB
时间: 2649ms
语言: Python3
提交时间: 2024-11-25 19:11:39

English 帮助 关于

sy316: 矩阵最大权值路径

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

思路:

在马走日的问题上稍作修改。深拷贝这块是问了ai才会用的。感觉自己不是很擅长处理这些细节上的语法。

代码:

```
def dfs(x,y,n,m,answer,remember,visited,chizu,count):
    if x==n-1 and y==m-1:
        answer.append(count[0])
        #remember.append([(0,0)])
        remember.append(remember[-1][:])
        return
    moves=[(1,0),(0,1),(0,-1),(-1,0)]
    for dx,dy in moves:
```

```
        nx, ny = x + dx, y + dy
        if 0<=nx<n and 0<=ny<m and not visited[nx][ny]:
            remember[-1].append((nx,ny))
            visited[nx][ny] = True
            count[0]+=chizu[nx][ny]
            dfs(nx,ny,n,m,answer,remember,visited,chizu,count)
            visited[nx][ny] = False
            count[0]-=chizu[nx][ny]
            remember[-1].pop()
n,m=map(int,input().split())
chizu=[]
for i in range(n):
    chizu.append(list(map(int,input().split())))
visited = [[False for _ in range(m)] for _ in range(n)]
visited[0][0]=True
remember=[[(0,0)]]
answer=[]
count=[chizu[0][0]]
dfs(0,0,n,m,answer,remember,visited,chizu,count)
max_quan=float('-inf')
true_remember=[]
for i in range(len(answer)):
    if max_quan<answer[i]:
        max_quan=answer[i]
        true_remember=remember[i]
for zuobiao in true_remember:
    print(zuobiao[0]+1,zuobiao[1]+1)
```

代码运行截图 （至少包含有"Accepted"）

代码书写 Python

```python
def dfs(x,y,n,m,answer,remember,visited,chizu,count):
    if x==n-1 and y==m-1:
        answer.append(count[0])
        #remember.append([(0,0)])
        remember.append(remember[-1][:])
        return
    moves=[(1,0),(0,1),(0,-1),(-1,0)]
    for dx,dy in moves:
        nx, ny = x + dx, y + dy
        if 0<=nx<n and 0<=ny<m and not visited[nx][ny]:
            remember[-1].append((nx,ny))
            visited[nx][ny] = True
            count[0]+=chizu[nx][ny]
            dfs(nx,ny,n,m,answer,remember,visited,chizu,cc
            visited[nx][ny] = False
            count[0]-=chizu[nx][ny]
            remember[-1].pop()
n,m=map(int,input().split())
chizu=[]
for i in range(n):
    chizu.append(list(map(int,input().split())))
```

测试输入 提交结果 历史提交

完美通过 查看题解

100% 数据通过测试

运行时长: 0 ms

LeetCode62.不同路径

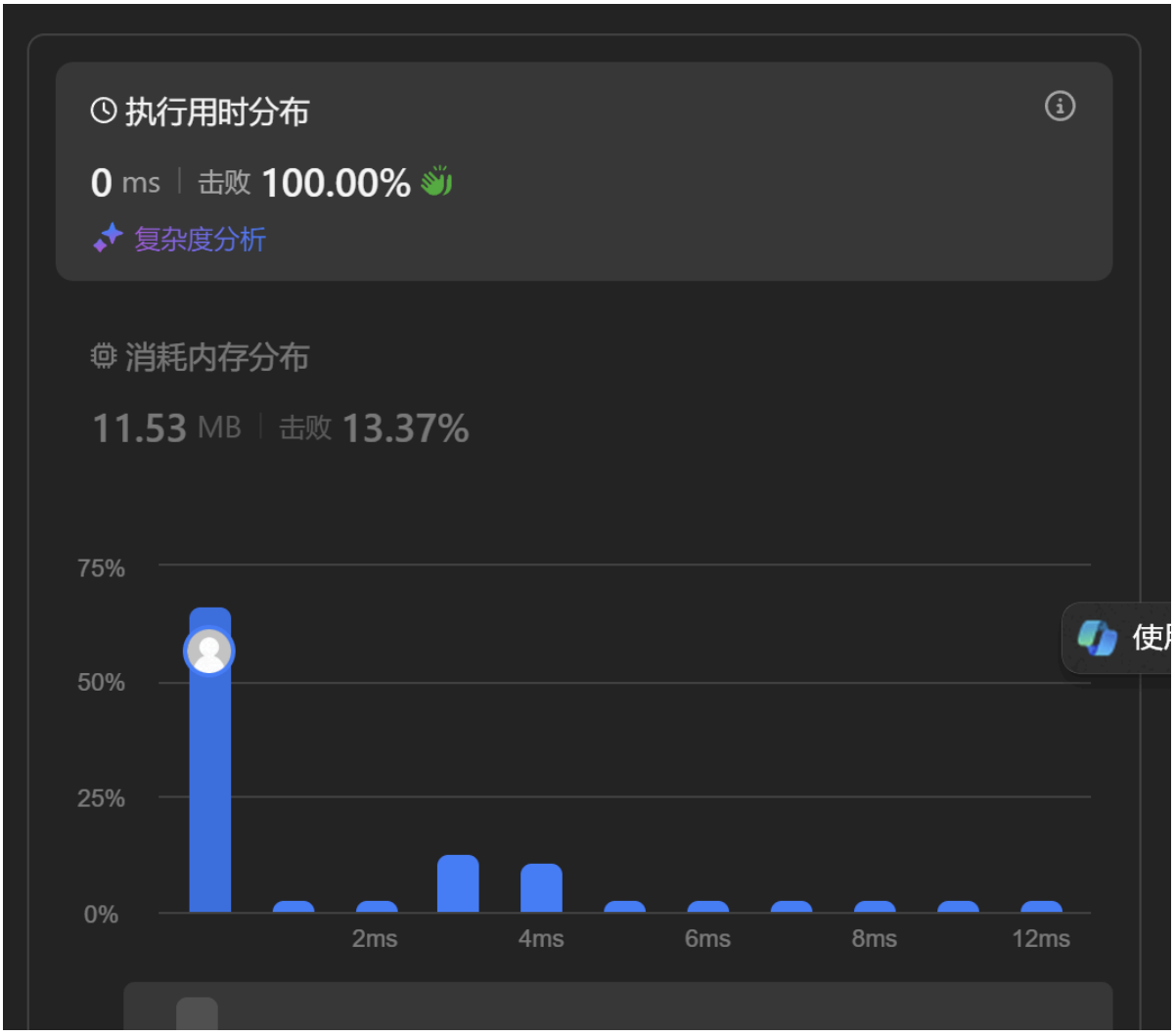
dp, https://leetcode.cn/problems/unique-paths/

思路：高中数学题，在总共m+n-2步中选出n-1步向下走，组合数

代码：

```
class Solution(object):
    def uniquePaths(self, m, n):
        def jie(n):
            result = 1
            for i in range(1, n+1):
                result *= i
            return result
        return(jie(m+n-2)/(jie(m-1)*jie(n-1)))
        """
        :type m: int
        :type n: int
        :rtype: int
        """
```

代码运行截图 （至少包含有"Accepted"）



sy358: 受到祝福的平方

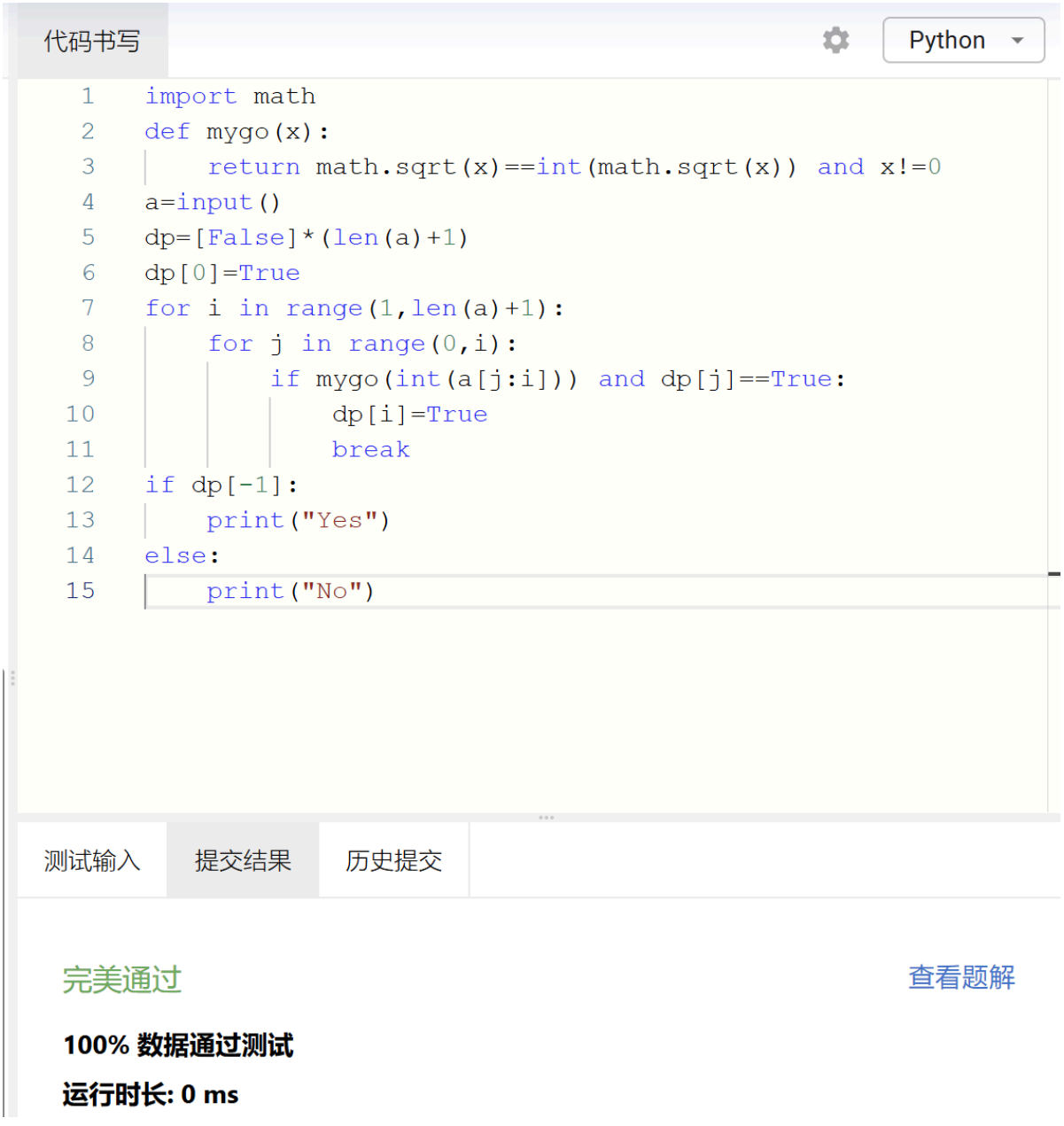
dfs, dp, https://sunnywhy.com/sfbj/8/3/539

思路：

代码：

```python
import math
def mygo(x):
    return math.sqrt(x)==int(math.sqrt(x)) and x!=0
a=input()
dp=[False]*(len(a)+1)
dp[0]=True
for i in range(1,len(a)+1):
    for j in range(0,i):
        if mygo(int(a[j:i])) and dp[j]==True:
            dp[i]=True
            break
if dp[-1]:
    print("Yes")
else:
    print("No")
```

代码运行截图（至少包含有"Accepted"）



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。

写了一万年，看到群里有人说这周作业太简单，悬着的心终于死了