Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by <mark>陶嘉瑞-物理学院</mark>

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：
使用 dfs 求出各个连通区域面积再取最大值。

代码：

```python
dir=[[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]
area=0

def dfs(x,y,matrix):
    global area
    if matrix[x][y]=='.':
        return
    matrix[x][y]='.'
    area+=1
    for i in range(len(dir)):
        dfs(x+dir[i][0],y+dir[i][1],matrix)

k=int(input())
```

```
for _ in range(k):
    n,m=map(int,input().split())
    matrix = [['.' for _ in range(m+2)] for _ in range(n+2)]
    for i in range(1,n+1):
        matrix[i][1:-1] = input()
    max_area=0
    for i in range(1,n+1):
        for j in range(1,m+1):
            if matrix[i][j]=='W':
                area=0
                dfs(i,j,matrix)
                max_area=max(max_area,area)
    print(max_area)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
dir=[[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]
area=0

def dfs(x,y,matrix):
    global area
    if matrix[x][y]=='.':
        return
    matrix[x][y]=' '
```

基本信息

#:	47299947
题目:	18160
提交人:	talenttao
内存:	3692kB
时间:	102ms
语言:	Python3
提交时间:	2024-11-21 10:30:13

19930: 寻宝

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

思路:
没想出来，参考答案的

代码:

```python
q = []

step = [[0, 1], [1, 0], [-1, 0], [0, -1]]
vis = [[0] * 52 for _ in range(52)]
g = []
```

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

def check(x, y):
    if (x < 0 or y < 0 or x >= m or y >= n):
        return False
    if (vis[x][y] or g[x][y] == 2):
        return False
    return True

q.append((0, 0))
head = 0
tail = 1
level = 0
while (head < tail):
    for k in range(head, tail):
        x, y = q[head]
        head += 1
        if (g[x][y] == 1):
            print(level)
            exit(0)
        for z in range(4):
            newx = x + step[z][0]
            newy = y + step[z][1]
            if (check(newx, newy)):
                vis[newx][newy] = 1
                q.append((newx, newy))
                tail += 1
    level += 1
print('NO')
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
q = []

step = [[0, 1], [1, 0], [-1, 0], [0, -1]]
vis = [[0] * 52 for _ in range(52)]
g = []

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

基本信息

#:	47302148
题目:	19930
提交人:	talenttao
内存:	3724kB
时间:	28ms
语言:	Python3
提交时间:	2024-11-21 13:14:27

04123: 马走日

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

思路:
利用递归即可

代码:

```python
k = 10
dirx = [-2,-1,1,2, 2, 1,-1,-2]
diry = [ 1, 2,2,1,-1,-2,-2,-1]
ans = 0

def dfs(num, x, y):
    if n*m == num:
        global ans
        ans += 1
        return
    for r in range(8):
        s = x + dirx[r]
        t = y + diry[r]
        if matrix[s][t]==False and 0<=s<n and 0<=t<m :
            matrix[s][t]=True
            dfs(num+1, s, t)
            matrix[s][t] = False

for _ in range(int(input())):
    n,m,x,y = map(int, input().split())
    matrix = [[False]*k for _ in range(k)]
    ans = 0
    matrix[x][y] = True
    dfs(1, x, y)
    print(ans)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
k = 10
dirx = [-2,-1,1,2, 2, 1,-1,-2]
diry = [ 1, 2,2,1,-1,-2,-2,-1]
ans = 0

def dfs(num, x, y):
    if n*m == num:
```

基本信息

#:	47300576
题目:	04123
提交人:	talenttao
内存:	3624kB
时间:	3332ms
语言:	Python3
提交时间:	2024-11-21 11:26:44

sy316: 矩阵最大权值路径

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

思路:
利用 dfs 求出每条可能的路径并取最大值

代码:

```python
def dfs(x,y,num):
    global max_num,best_path
    if x==n-1 and y==m-1:
        if num>max_num:
            max_num=num
            best_path=path[:]
        return
    visited[x][y]=True
    for i in range(4):
        x1=x+dx[i]
        y1=y+dy[i]
        if 0<=x1<n and 0<=y1<m and (not visited[x1][y1]):
            num1=num+matrix[x1][y1]
            path.append((x1,y1))
            dfs(x1,y1,num1)
            path.pop()
    visited[x][y]=False


n,m=map(int,input().split())
matrix=[]
for i in range(n):
    matrix.append(list(map(int,input().split())))
visited=[[False for i in range(m)] for j in range(n)]
```

```
dx=[0,0,1,-1]
dy=[1,-1,0,0]
max_num=-float('inf')
best_path=[]
path=[(0,0)]
dfs(0,0,matrix[0][0])
for x,y in best_path:
    print(x+1,y+1)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

LeetCode62.不同路径

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

思路:
可用数学方法直接计算

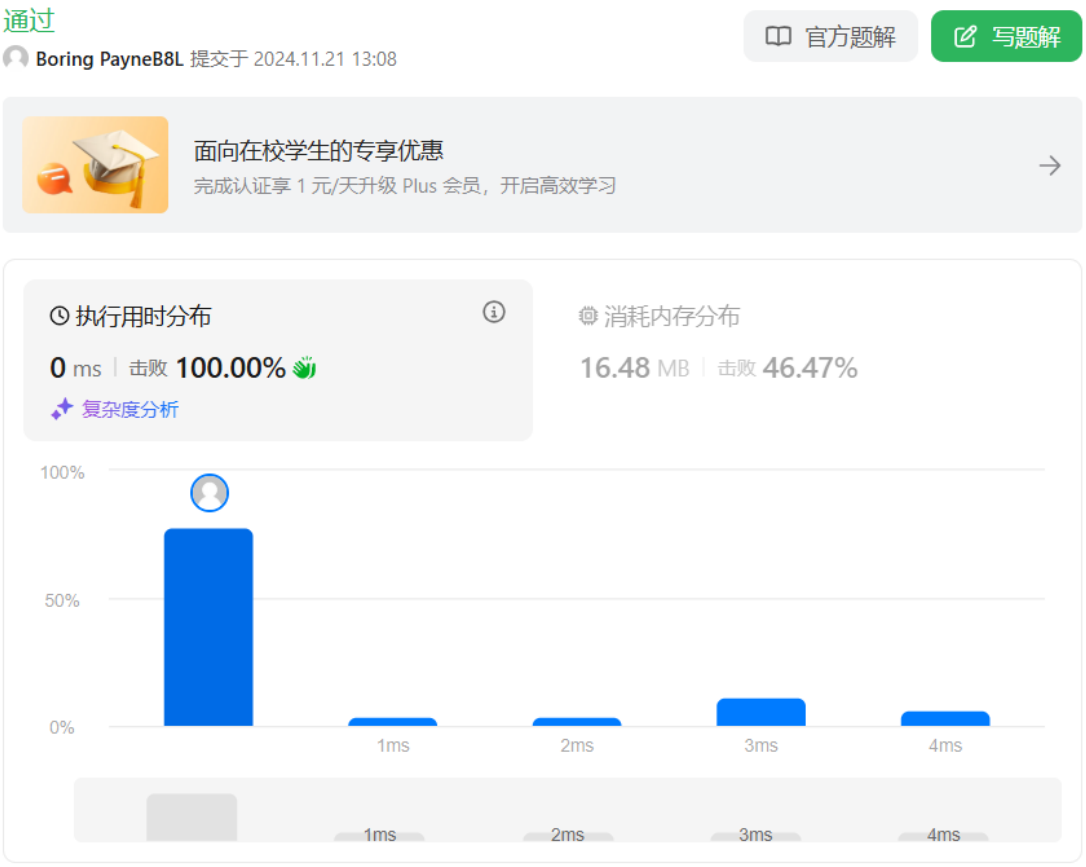
代码:

```python
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        array=[1]*(n+m-1)
```

```
        for i in range(1,n+m-1):
            array[i]=i*array[i-1]
        return (array[-1]//(array[n-1]*array[m-1]))
```

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>




### sy358: 受到祝福的平方

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

思路:
使用 dfs 暴力遍历

代码:

```python
```
```

```
squares = set()
i = 1
while i * i <= 10**9:
    squares.add(str(i * i))
    i += 1

def dfs(num):
    global judge
    for i in range(1,len(num)+1):
        num1=num[:i]
        if num1 in squares:
            if num1==num:
                judge=True
            else:
                num2=num[i:]
                dfs(num2)



num=input()
judge=False
dfs(num)
print('Yes' if judge else 'No')

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

受到祝福的平方

通过数 183 提交数 387 难度 中等 显示标签

题目描述

在小元的世界里，任何人出生后会被世界分配一个随机ID，如果ID在被切割后，即ID满足按照从左至右顺序分割，且分割出来的数字都是某一个 正整数 的平方，分割时可以包括前导0，那么他就被这个世界祝福，最后获得快乐的数量和质量都比不满足这样的ID的人多的多。

令ID为A，且A是一个正整数，取值范围为$1 \leq A \leq 10^9$，问A是否是一个被受到祝福的ID。

比如$A = 8194$时，它是一个被受到祝福的ID，因为他可以被分割为$\{81, 9, 4\} = \{9^2, 3^2, 2^2\}$；

比如$A = 1001$时，它是一个被受到祝福的ID，因为他可以被分割为$\{1, 001\} = \{1^2, 1^2\}$，或者$\{100, 1\} = \{10^2, 1^2\}$。注意$\{1, 00, 1\} = \{1^2, 0^2, 1^2\}$不是一个合法切割，因为分割出来的数字必须为正整数的平方；

比如$A = 36$时，36已经是一个平方数了，所以它同样满足条件；

```
while i * i <= 10**9:
    squares.add(str(i * i))
    i += 1

def dfs(num):
    global judge
    for i in range(1,len(num)+1):
        num1=num[:i]
        if num1 in squares:
            if num1==num:
                judge=True
            else:
                num2=num[i:]
                dfs(num2)

num=input()
```

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

2. 学习总结和收获

<mark>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。</mark>
本周在补之前的练习，感觉 dp 和 dfs 以及掌握的较好，但 bfs 还不太会，作业中的 bfs 题目想了很久没想出来最后还是参考的答案，感觉还需要投入更多的精力。