

Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Compiled by 陈俊逸、工院

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 (25min)

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

思路:

有点像上课讲的例题，先求每块岛屿的w数，记为judge，最后答案是最大的judge

代码:

```
directions=[[-1,1],[0,1],[1,1],[-1,-1],[1,-1],[0,-1],[-1,0],[1,0]]
def dfs(ma,i,j,ans):
    for dx,dy in directions:
        if ma[i+dx][j+dy]=='w':
            ans+=1
            ma[i + dx][j + dy] = '.'
            ma,ans=dfs(ma,i+dx,j+dy,ans)
    return ma,ans
case=int(input())
for _ in range(case):
    n,m=map(int,input().split())
    ma=[]
    ans,judge=0,0
    ma.append(['.' for _ in range(m+2)])
    for _ in range(n):
        ma.append(['.']+list(input())+['.'])
    ma.append(['.' for _ in range(m+2)])
    for i in range(1,n+1):
        for j in range(1,m+1):
            if ma[i][j]=='w':
                ma[i][j]='.'
                ma,judge=dfs(ma,i,j,1)
                ans=judge if judge>ans else ans
```

```
print(ans)
```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011257 信箱 账号

 **CS101 / 题库 (包括计概、数算题目)**

[题目](#) [排名](#) [状态](#) [提问](#)

#47332155提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
directions=[[-1,1],[0,1],[1,1],[-1,-1],[1,-1],[0,-1],[-1,0],[1,0]]
def dfs (ma,i,j,ans):
    for dx,dy in directions:
        if ma[i+dx][j+dy]=='W':
            ans+=1
            ma[i + dx][j + dy] = '.'
            ma,ans=dfs (ma,i+dx,j+dy,ans)
    return ma,ans
case=int(input())
for _ in range(case):
    n,m=map(int,input().split())
    ma=[]
    ans,judge=0,0
    ma.append(['.' for _ in range(m+2)])
    for _ in range(n):
```

基本信息

#: 47332155

题目: 18160

提交人: 24n2400011257

内存: 3744kB

时间: 73ms

语言: Python3

提交时间: 2024-11-22 18:37:58

19930: 寻宝(30min)

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

思路:

记录已经走过的位置, 选择可以走的位置不断bfs

代码:

```
directions=[[1,0],[-1,0],[0,-1],[0,1]]
def main():
    n,m=map(int,input().split())
    ma=[]
    ma.append([2 for _ in range(m+2)])
    for i in range(1,n+1):
        ma.append([2]+list(map(int,input().split()))+[2])
    ma.append([2 for _ in range(m+2)])
    print(bfs(ma,1,1))
def bfs(ma,i,j):
    stack=[(i,j,0)]
    visited=set([(i,j)])
    while stack:
        x,y,step=stack.pop(0)
        if ma[x][y]==1:
            return step
        for dx,dy in directions:
            if (x+dx,y+dy) not in visited and ma[x+dx][y+dy]!=2:
                stack.append((x+dx,y+dy,step+1))
                visited.add((x+dx,y+dy))
    return 'NO'
main()
```

代码运行截图 == (至少包含有"Accepted") ==

OpenJudge

题目ID, 标题, 描述

24n2400011257

信箱

账号

CS101 / 题库 (包括计概、数算题目)

题目

排名

状态

提问

#47333274提交状态

查看

提交

统计

提问

状态: Accepted

基本信息

#: 47333274

题目: 19930

提交人: 24n2400011257

内存: 3696kB

时间: 29ms

语言: Python3

提交时间: 2024-11-22 19:27:52

源代码

```
directions=[[1,0],[-1,0],[0,-1],[0,1]]
def main():
    n,m=map(int,input().split())
    ma=[]
    ma.append([2 for _ in range(m+2)])
    for i in range(1,n+1):
        ma.append([2]+list(map(int,input().split()))+[2])
    ma.append([2 for _ in range(m+2)])
    print(bfs(ma,1,1))
def bfs(ma,i,j):
    stack=[(i,j,0)]
    visited=set([(i,j)])
    while stack:
        x,y,step=stack.pop(0)
        if ma[x][y]==1:
```

04123: 马走日

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

思路:

先尽可能走完一种, 然后回溯, 看有无其他方法, 以此类推: 不断回溯, 尽可能走, 回溯, 尽可能走。

代码:

```
directions=[[-2,1],[-1,2],[1,2],[2,1],[2,-1],[1,-2],[-1,-2],[-2,-1]]
def dfs(n,m,x,y,ma,step,p):
    if step==n*m:
        p[0]+=1
        return
    for dx,dy in directions:
        if -1<x+dx<n and -1<y+dy<m and ma[x+dx][y+dy]:
            ma[x+dx][y+dy]=0
            dfs(n,m,x+dx,y+dy,ma,step+1,p)
            ma[x+dx][y+dy]=1
case=int(input())
for _ in range(case):
    n,m,x,y=map(int,input().split())
    ma=[[1 for _ in range(m)] for _ in range(n)]
    ma[x][y]=0
    p=[0]
    dfs(n,m,x,y,ma,1,p)
    print(p[0])
```

代码运行截图 (至少包含有"Accepted")

CS101 / 题单 (已订时佩、双开题单)

题目 排名 状态 提问

#47334456提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
directions=[[-2,1],[-1,2],[1,2],[2,1],[2,-1],[1,-2],[-1,-2],[-2,-1]]
def dfs(n,m,x,y,ma,step,p):
    if step==n*m:
        p[0]+=1
        return
    for dx,dy in directions:
        if -1<x+dx<n and -1<y+dy<m and ma[x+dx][y+dy]:
            ma[x+dx][y+dy]=0
            dfs(n,m,x+dx,y+dy,ma,step+1,p)
            ma[x+dx][y+dy]=1
case=int(input())
for _ in range(case):
    n,m,x,y=map(int,input().split())
    ma=[[1 for _ in range(m)] for _ in range(n)]
    ma[x][y]=0
    p=[0]
    dfs(n,m,x,y,ma,1,p)
    print(p[0])
```

基本信息

#: 47334456

题目: 04123

提交人: 24n2400011257

内存: 3664kB

时间: 2764ms

语言: Python3

提交时间: 2024-11-22 20:15:54

sy316: 矩阵最大权值路径 (int long long time)

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

思路:

dfs找每个路径的和, 比较全局变量zuidazhi和he的大小, 用visited记录点位

代码:

```
import copy
def dfs(x,y,visited,he):
    global n,m,ma,zuidazhi,v
    if x==n-1 and y==m-1:
        if zuidazhi<he+ma[-1][-1]:
            zuidazhi=he+ma[-1][-1]
            #print(he+ma[-1][-1])
            v=copy.copy(visited)
        return
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if -1<nx<n and -1<ny<m and (nx,ny) not in visited:
            visited.append((nx,ny))
            dfs(nx,ny,visited,he+ma[x][y])
            visited.remove((nx,ny))
    return
n,m=map(int,input().split())
ma=[]
directions=[(1,0),(0,1),(-1,0),(0,-1)]
zuidazhi=float('-inf')
v=[]
for _ in range(n):
    ma.append(list(map(int,input().split())))
dfs(0,0,[(0,0)],0)
for i,j in v:
    print(i+1,j+1)
```

代码运行截图 (至少包含有"Accepted")



LeetCode62.不同路径 (5min)

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

思路:

小学就会的一个数学方法

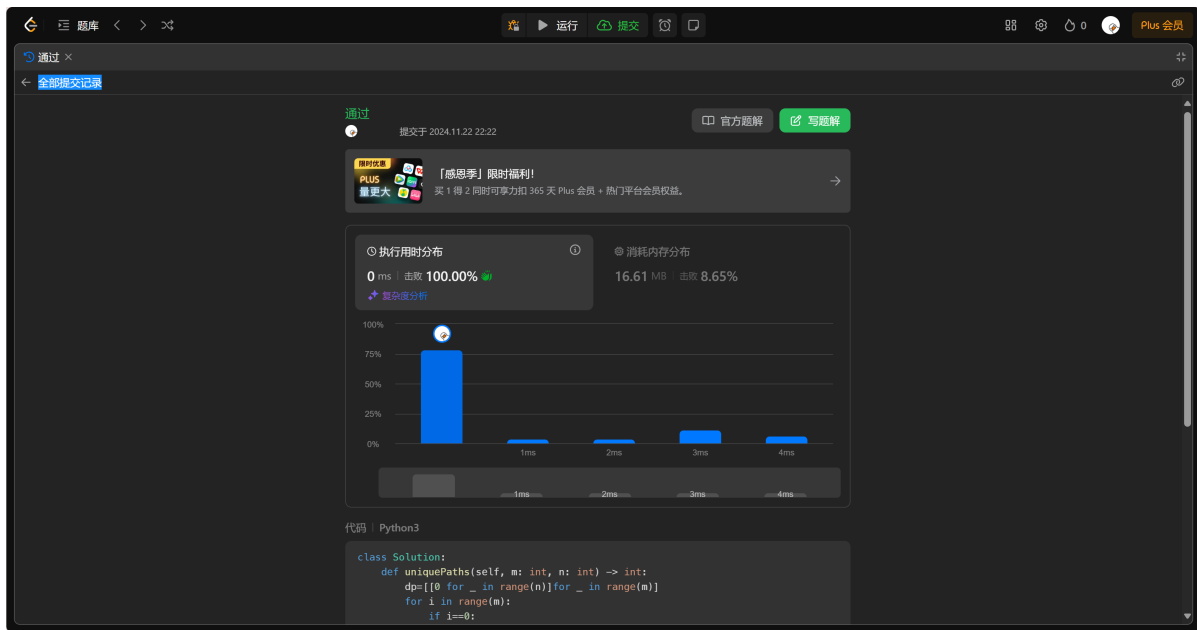
代码:

```

class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        dp=[[0 for _ in range(n)]for _ in range(m)]
        for i in range(m):
            if i==0:
                for j in range(n):
                    dp[0][j]=1
                for j in range(m):
                    dp[j][0]=1
            else:
                for j in range(n):
                    if not dp[i][j]:
                        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")



sy358: 受到祝福的平方 (15min)

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

思路：先打一个表，然后判断切下来的数在不在里头。如果有能切的方法，在j里面加一个东西，然后就可以输出yes了

代码：

```
def cut(id):
    n=len(id)
    if n==0:
        j.append('oooooooo')
        return
    for i in range(n):
        if int(id[0:i+1]) in judge:
            iid=id[i+1:]
            cut(iid)
    return
id=str(input())
judge=[x**2 for x in range(1,31622)]
judge=set(judge)
j=[]
cut(id)
print('No' if not j else 'Yes')
```

代码运行截图 (至少包含有"Accepted")

晴问

课程 训练营 算法笔记 题库 比赛 语言入门教程 考研算法大题特训 新增功能 本期速递

提高篇 (2) — 搜索专题

综合练习精选

受到祝福的平方

题目 题解

样例1

输入 复制

8194

输出 复制

Yes

样例2

输入 复制

3

输出 复制

No

样例3

输入 复制

代码书写

```
1 def cut(id):
2     n=len(id)
3     if n==0:
4         j.append('oooooooo')
5         return
6     for i in range(n):
7         if int(id[0:i+1]) in judge:
8             iid=id[i+1:]
9             切割 (iid)
10            return
11 id=str(input())
12 judge=[x**2 for x in range(1,31622)]
13 judge=set(judge)
14 j=[]
15 cut(id)
16 print('No' if not j else 'Yes')
```

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

收起面板 运行 提交

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。

刚刚接触搜索，感觉还是不太熟练，前几题能写个大概，但都有小问题，最后通过ai修了修就过了。

后面几个题在室友大佬的指导下，慢慢对dfs有感觉了，也基本搞懂了bfs。感觉还是得稍微刷一点题，练练手。

这周的题有些还是比较有意思的，比如倒数第二个题，做的时候感觉回到了我小学学奥数的时候。我给我室友说我不用dfs也能做出来，他当时还不信hh。（后面也用dfs写了一遍，就当练练手，很简单，感觉这个题应该放在第一题的位置）

矩阵呢个题感觉还是有新收获，就是发现了v=visited和v=copy.copy(visited)的不同，感觉很神奇。通过ai也有个大概的了解。