Assignment #9: 图论：遍历，及 树算

Updated 1739 GMT+8 Apr 14, 2024

2024 spring, Complied by 李鹏辉，元培学院

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。

编程环境

Windows 10 Home, PyCharm 2022.3.2 (Community Edition)

操作系统：macOS Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

1. 题目

04081: 树的转换

http://cs101.openjudge.cn/dsapre/04081/

思路：about 20 mins.

代码

```
class Tree:
    def __init__(self, node):
        self.node = node
        self.children = []
        self.parent = None
        self.depth = 0
        self.new_depth = 0


def q1():
    dus = input()
```

```
    t = Tree(0)
    c_node = t  # current_node
    c_number = 0
    m_d_b = 0  # max_depth_before
    for _, i in enumerate(dus):
        if i == 'd':
            c_number += 1
            c_node.children.append(Tree(c_number))
            c_node.children[-1].parent = c_node
            c_node.children[-1].depth = c_node.depth + 1
            c_node = c_node.children[-1]
            if c_node.depth > m_d_b:
                m_d_b = c_node.depth
        else:
            c_node = c_node.parent
    q = [t]
    m_d_a = 0  # max_depth_after
    while q:
        root = q.pop(0)
        for i, n in enumerate(root.children):
            n.new_depth = root.new_depth + i + 1
            if n.new_depth > m_d_a:
                m_d_a = n.new_depth
            q.append(n)
    print(f'{m_d_b} => {m_d_a}')


q1()
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
class Tree:
    def __init__(self, node):
        self.node = node
        self.children = []
        self.parent = None
        self.depth = 0
```

基本信息

#: 44762303
题目: 04081
提交人: 2100017777
内存: 3680kB
时间: 29ms
语言: Python3
提交时间: 2024-04-23 13:51:14

08581: 扩展二叉树

http://cs101.openjudge.cn/dsapre/08581/

思路：about 20 mins.

代码

```
class BTree:
    def __init__(self, node):
        self.node = node
```

```
        self.left = None
        self.right = None


def q2():
    def build_tree(r):
        cs = r[1:]  # current string
        t = BTree(r[0])
        s = [t]
        while cs:
            nt = BTree(cs[0])  # new tree
            cs = cs[1:]
            if s[-1].left is None:
                s[-1].left = nt
                if nt.node != '.':
                    s.append(nt)
            else:
                s[-1].right = nt
                s.pop()
                if nt.node != '.':
                    s.append(nt)
        return t

    def inorder(t):
        re = ''
        if t.left.node != '.':
            re += inorder(t.left)
        re += t.node
        if t.right.node != '.':
            re += inorder(t.right)
        return re

    def postorder(t):
        re = ''
        if t.left.node != '.':
            re += postorder(t.left)
        if t.right.node != '.':
            re += postorder(t.right)
        re += t.node
        return re

    t = build_tree(input())
    print(inorder(t))
    print(postorder(t))


q2()
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
class BTree:
    def __init__(self, node):
        self.node = node
        self.left = None
        self.right = None
```

基本信息

#:	44762488
题目:	08581
提交人:	2100017777
内存:	3684kB
时间:	27ms
语言:	Python3
提交时间:	2024-04-23 14:14:41

22067: 快速堆猪

http://cs101.openjudge.cn/practice/22067/

思路：about 25 mins.

代码

```
def q3():
    ms = []  # min stack
    cm = -1  # current min
    ws = []  # weight stack
    try:
        while True:
            o = input()
            if o == 'min':
                if cm == -1:
                    continue
                else:
                    print(cm)

            if o.startswith('push'):
                _, w = o.split()
                w = int(w)
                ws.append(w)
                if w <= cm or cm == -1:
                    cm = w
                    ms.append(w)

            if o == 'pop':
                if cm == -1:
                    continue
                else:
                    top = ws.pop()
                    if top == cm:
                        ms.pop()
                        if ms:
                            cm = ms[-1]
                        else:
                            cm = -1

    except EOFError:
        return
```

```
36
37
38 q3()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

```
def q3():
    ms = []  # min stack
    cm = -1  # current min
    ws = []  # weight stack
    try:
        while True:
```

基本信息

#:	44762640
题目:	22067
提交人:	2100017777
内存:	6720kB
时间:	288ms
语言:	Python3
提交时间:	2024-04-23 14:39:37

04123: 马走日

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

思路：about 30 mins.

代码

```
1  class Point:
2      def __init__(self, ors):
3          self.ors = ors
4          self.suns = []
5          self.av = True
6
7
8  def q4():
9      def build_graph(n, m):
10         pd = {}  # points_dict
11         for i in range(0, n):
12             for j in range(0, m):
13                 pd[(i, j)] = Point((i, j))
14
15         dirs = [(1, 2), (2, 1), (1, -2), (2, -1), (-1, -2), (-2, -1), (-1,
   2), (-2, 1)]
16         for i in range(0, n):
17             for j in range(0, m):
18                 suns = []
19                 for direction in dirs:
20                     xm, ym = direction
21                     if 0 <= i+xm <= n-1 and 0 <= j+ym <= m-1:
22                         suns.append((i+xm, j+ym))
23                 pd[(i, j)].suns.extend([pd[sun] for sun in suns])
24         return pd
25
26     for _ in range(int(input())):
27         il = map(int, input().split())  # input split
```

```
        n, m, x, y = il
        pd = build_graph(n, m)
        fl = n * m  # full length
        apps = 0

        def dfs(ap, pl):  # aim point, past length
            nonlocal apps
            if pl == fl - 1:
                apps += 1
            else:
                ap.av = False
                for ps in ap.suns:  # possible 'sun'
                    if ps.av:
                        dfs(ps, pl+1)
                ap.av = True

        dfs(pd[(x, y)], 0)
        print(apps)


q4()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态：Accepted

源代码

```
class Point:
    def __init__(self, ors):
        self.ors = ors
        self.suns = []
        self.av = True
```

基本信息

#: 44762877
题目: 04123
提交人: 2100017777
内存: 7456kB
时间: 695ms
语言: Python3
提交时间: 2024-04-23 15:13:16

28046: 词梯

bfs, http://cs101.openjudge.cn/practice/28046/

思路：about 40 mins.

代码

```
class Word:
    def __init__(self, word):
        self.node = word
        self.nbrs = []
        self.av = True
        self.prev = None


def q5():
    def build_graph():
```

```
        wd = {}  # word dict
        for _ in range(int(input())):
            w = input()
            wd[w] = Word(w)

        wmd = {}  # word model dict
        for w in wd.keys():
            for i in range(4):
                m = w[:i] + '_' + w[i+1:]  # model
                if m in wmd:
                    wmd[m].append(wd[w])
                else:
                    wmd[m] = [wd[w]]
        for m in wmd.values():
            for w in m:
                for ow in m:  # other words
                    if ow != w:
                        w.nbrs.append(ow)
        return wd

    wd = build_graph()
    start, des = input().split()
    if start not in wd.keys() or des not in wd.keys():
        print('NO')
        return
    wd[start].av = False
    q = [wd[start]]

    while q:
        aw = q.pop(0)
        if aw.node == des:
            rl = [des]  # result list
            cw = aw.prev  # current word
            while cw is not None:
                rl.append(cw.node)
                cw = cw.prev
            print(' '.join(rl[::-1]))
            return
        else:
            for w in aw.nbrs:
                if w.av:
                    w.av = False
                    w.prev = aw
                    q.append(w)
    print('NO')
    return


q5()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

```
class Word:
    def __init__(self, word):
        self.node = word
        self.nbrs = []
        self.av = True
        self.prev = None
```

基本信息

#: 44763430
题目: 28046
提交人: 2100017777
内存: 6896kB
时间: 53ms
语言: Python3
提交时间: 2024-04-23 15:56:21

28050: 骑士周游

dfs, http://cs101.openjudge.cn/practice/28050/

思路：about 12 mins, based on question 4.

代码

```
import sys


class Point:
    def __init__(self, ors):
        self.ors = ors
        self.suns = []
        self.av = True


def q6():
    def build_graph(n, m):
        pd = {}  # points_dict
        for i in range(0, n):
            for j in range(0, m):
                pd[(i, j)] = Point((i, j))

        dirs = [(1, 2), (2, 1), (1, -2), (2, -1), (-1, -2), (-2, -1), (-1, 2), (-2, 1)]
        for i in range(0, n):
            for j in range(0, m):
                suns = []
                for direction in dirs:
                    xm, ym = direction
                    if 0 <= i+xm <= n-1 and 0 <= j+ym <= m-1:
                        suns.append((i+xm, j+ym))
                pd[(i, j)].suns.extend([pd[sun] for sun in suns])
        return pd

    n = int(input())
    il = map(int, input().split())  # input split
    x, y = il
    pd = build_graph(n, n)
    fl = n * n  # full length

```

```
    def dfs(ap, pl):  # aim point, past length
        if pl == fl - 1:
            print('success')
            sys.exit(0)
        else:
            ap.av = False
            ps = []
            for s in ap.suns:  # possible 'sun'
                if s.av:
                    sps = 0  # the sun's possible suns
                    for ss in s.suns:  # suns of the sun
                        if ss.av:
                            sps += 1
                    ps.append([s, sps])
            ps.sort(key=lambda x: x[1])
            for s in ps:
                dfs(s[0], pl+1)
            ap.av = True

    dfs(pd[(x, y)], 0)
    print('fail')


q6()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态：Accepted

源代码

```
import sys


class Point:
    def __init__(self, ors):
        self.ors = ors
```

基本信息

#:	44763650
题目:	28050
提交人:	2100017777
内存:	4224kB
时间:	30ms
语言:	Python3
提交时间:	2024-04-23 16:12:28

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“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。

图的bfs与dfs有一定差别，一是标颜色的时机，二是bfs用队列，而dfs用函数递归而非栈。