

Assignment #9: 图论：遍历，及 树算

Updated 1739 GMT+8 Apr 14, 2024

2024 spring, Complied by 田济维 物理学院

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

(python pycharm)

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：

代码

```
1  #
2  class TreeNode:
3      def __init__(self, parent=None):
4          self.children = []
5          self.parent = parent
6
7  def buildtree(seq):
8      maxn = 0
9      height = 0
```

```

10     root = TreeNode()
11     currentnode = root
12     for x in seq:
13         if x == "d":
14             height +=1
15             maxn = max(maxn,height)
16             s = TreeNode(currentnode)
17             currentnode.children.append(s)
18             currentnode = s
19         if x == "u":
20             height-=1
21             maxn = max(maxn,height)
22             currentnode = currentnode.parent
23     return [root,maxn]
24
25 def countheight(root):
26     if root.children:
27         depth = [countheight(root.children[i])+1+i for i in
28 range(len(root.children))]
29         return max(depth)
30     else:
31         return 0
32
33 s = input()
34 l = buildtree(s)
35 root = l[0]
36 h1 = l[1]
37 h2 = countheight(root)
38 print(f"{h1} => {h2}")

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

基本信

源代码

```

class TreeNode:
    def __init__(self,parent=None):
        self.children = []
        self.parent = parent

def buildtree(seq):
    maxn = 0
    height = 0
    root = TreeNode()
    currentnode = root
    for x in seq:
        if x == "d":
            height +=1
            maxn = max(maxn,height)
            s = TreeNode(currentnode)
            currentnode.children.append(s)
            currentnode = s
        if x == "u":
            height-=1
            maxn = max(maxn,height)
            currentnode = currentnode.parent

```

提

提交

08581: 扩展二叉树

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

思路:

代码

```
1  #
2  class TreeNode:
3      def __init__(self, key, parent=None):
4          self.key = key
5          self.parent = parent
6          self.left = None
7          self.right = None
8
9  def buildTree(seq):
10     s = TreeNode(seq[0], None)
11     current = s
12     for x in seq[1:]:
13         temp = TreeNode(x, current)
14         if current.left == None:
15             current.left = temp
16         else:
17             current.right = temp
18             current = temp
19     while (current.left != None and current.right != None) or current.key
== ".":
20         current = current.parent
21         if current == None:
22             break
23     return s
24
25 def inorder(tree):
26     if tree.key != ".":
27         inorder(tree.left)
28         print(tree.key, end = "")
29         inorder(tree.right)
30
31 def postorder(tree):
32     if tree.key != ".":
33         postorder(tree.left)
34         postorder(tree.right)
35         print(tree.key, end="")
36
37 s = input()
38 tree = buildTree(s)
39
40 inorder(tree)
41 print("")
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
class TreeNode:
    def __init__(self, key, parent=None):
        self.key = key
        self.parent = parent
        self.left = None
        self.right = None

    def buildTree(seq):
        s = TreeNode(seq[0], None)
        current = s
        for x in seq[1:]:
            temp = TreeNode(x, current)
            if current.left == None:
                current.left = temp
            else:
                current.right = temp
            current = temp
        while (current.left != None and current.right != None) or current.left == None:
            current = current.parent
            if current == None:
                break
        return s

    def inorder(tree):
        if tree.key != ".":
            inorder(tree.left)
            print(tree.key, end = " ")
```

基本信息

#

题目

提交人

内存

时间

语言

提交时间

22067: 快速堆猪

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

思路:

代码

```
1 #
2 stack = []
3 minstack = [1000000]
4 while True:
5     try:
6         s = input().split()
7     except EOFError:
8         break
9     else:
10         if s[0] == "push":
11             stack.append(int(s[1]))
```

```

12         minstack.append(min(int(s[1]),minstack[-1]))
13     elif s[0] == "pop":
14         if stack:
15             stack.pop()
16             minstack.pop()
17     else:
18         if stack:
19             print(minstack[-1])
20

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```

stack = []
minstack = [1000000]
while True:
    try:
        s = input().split()
    except EOFError:
        break
    else:
        if s[0] == "push":
            stack.append(int(s[1]))
            minstack.append(min(int(s[1]),minstack[-1]))
        elif s[0] == "pop":
            if stack:
                stack.pop()
                minstack.pop()
        else:
            if stack:
                print(minstack[-1])

```

04123: 马走日

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

思路:

代码

```

1  #
2  sx = [2,1,-1,-2,-2,-1,1,2]
3  sy = [1,2,2,1,-1,-2,-2,-1]
4  ans = [0]
5  def dfs(seq,x,y):
6

```

```

7     if seq == n*m:
8         ans[0]+=1
9         return
10    for i in range(8):
11        tx = x+sx[i]
12        ty = y+sy[i]
13        if not (0<=tx<n and 0<=ty<m):
14            continue
15        elif M[tx][ty]== False:
16            M[tx][ty]=True
17            dfs(seq+1,tx,ty)
18            M[tx][ty]=False
19
20
21
22    T = int(input())
23    for _ in range(T):
24        n, m, x, y = map(int,input().split())
25        ans[0] = 0
26        M = [[False]*m for i in range(n)]
27        M[x][y] = True
28        dfs(1,x,y)
29        print(ans[0])

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

基

源代码

```

sx = [2,1,-1,-2,-2,-1,1,2]
sy = [1,2,2,1,-1,-2,-2,-1]
ans = [0]
def dfs(seq,x,y):

    if seq == n*m:
        ans[0]+=1
        return
    for i in range(8):
        tx = x+sx[i]
        ty = y+sy[i]
        if not (0<=tx<n and 0<=ty<m):
            continue
        elif M[tx][ty]== False:
            M[tx][ty]=True
            dfs(seq+1,tx,ty)
            M[tx][ty]=False

T = int(input())
for _ in range(T):

```

基

28046: 词梯

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

思路:

代码

```
1  #
2  from collections import deque
3  n = int(input())
4  buckets = {}
5  for _ in range(n):
6      s = input()
7      for i in range(4):
8          bucket = f"{s[:i]}_{s[i+1:]}"
9          if bucket not in buckets:
10             buckets[bucket]=[s]
11         else:
12             buckets[bucket].append(s)
13
14  def bfs(s,e):
15      que = deque([[s]])
16      visited = {s:1}
17      while que:
18          t = que.popleft()
19          word = t[-1]
20          for i in range(4):
21              bucket = f"{word[:i]}_{word[i+1:]}"
22              for x in buckets[bucket]:
23                  if x == e:
24                      return t
25                  elif x not in visited:
26                      que.append(t+[x])
27                      visited[x]=1
28      return "NO"
29  s,e = input().split()
30  road = bfs(s,e)
31  if road == "NO":
32      print(road)
33      exit()
34  road.append(e)
35  print(" ".join(road))
36
37
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

基本信息

源代码

```
from collections import deque
n = int(input())
buckets = {}
for _ in range(n):
    s = input()
    for i in range(4):
        bucket = f"{s[:i]}_{s[i+1:]}"
        if bucket not in buckets:
            buckets[bucket] = [s]
        else:
            buckets[bucket].append(s)

def bfs(s, e):
    que = deque([[s]])
    visited = {s:1}
    while que:
        t = que.popleft()
        word = t[-1]
        for i in range(4):
            bucket = f"{word[:i]}_{word[i+1:]}"
```

#

题目

提交人

内存

时间

语言

提交时间

28050: 骑士周游

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

思路:

代码

```
1  #
2  n = int(input())
3  sx = [2,1,-1,-2,-2,-1,1,2]
4  sy = [1,2,2,1,-1,-2,-2,-1]
5  def postoid(i,j,n):
6      return i*n+j
7
8  def buildgraph(n):
9      for x in range(n):
10         for y in range(n):
11             node_id = postoid(x,y,n)
12             for i in range(8):
13                 tx = x+sx[i]
14                 ty = y+sy[i]
15                 if 0<=tx<n and 0<=ty<n:
16                     anotherid = postoid(tx,ty,n)
17                     graph[node_id].append(anotherid)
18
19
20 def sortnode(u):
21     s = graph[u]
22     eff = []
23     for x in s:
```



```

24         if Color[x]:
25             w = 0
26             for y in graph[x]:
27                 if Color[y]:
28                     w+=1
29             eff.append((x,w))
30     temp = sorted(eff,key = lambda x:x[1])
31     return temp
32
33 def dfs(depth,u,n):
34     Color[u]=False
35     if depth == n*n:
36         Color[u]=True
37         return True
38
39     g = sortnode(u)
40     for x in g:
41         if Color[x[0]] and dfs(depth+1,x[0],n):
42             Color[u]=True
43             return True
44     Color[u]=True
45     return False
46
47
48 graph = {}
49 x,y = map(int,input().split())
50 for i in range(n*n):
51     graph[i]=[]
52 buildgraph(n)
53 Color = [True for i in range(n * n)]
54 result = dfs(1,postoid(x,y,n),n)
55 if result:
56     print("success")
57 else:
58     print("fail")
59

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
n = int(input())
sx = [2,1,-1,-2,-2,-1,1,2]
sy = [1,2,2,1,-1,-2,-2,-1]
def postoid(i,j,n):
    return i*n+j

def buildgraph(n):
    for x in range(n):
        for y in range(n):
            node_id = postoid(x,y,n)
            for i in range(8):
                tx = x+sx[i]
                ty = y+sy[i]
                if 0<=tx<n and 0<=ty<n:
                    anotherid = postoid(tx,ty,n)
                    graph[node_id].append(anotherid)

def sortnode(u):
    s = graph[u]
    eff = []
```

基本信息

#: 44760863
题目: 28050
提交人: 23n2300011503
内存: 3996kB
时间: 28ms
语言: Python3
提交时间: 2024-04-23 11:23:13

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。

感觉做作业时被第一题搞心态了（一般觉得作业第一题很简单），整体来说本次作业难度不高。花了3h