

# Assignment #9: 图论：遍历，及树算

---

Updated 1739 GMT+8 Apr 14, 2024

2024 spring, Compiled by 钟明衡 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://typora.io.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

操作系统：Windows\_NT x64 10.0.19045

Python编程环境：Visual Studio Code 1.76.1

C/C++编程环境：Visual Studio Code 1.76.1

## 1. 题目

---

### 04081: 树的转换

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

思路：

用一个指标h来指示当前处于原树的第几层，而利用列表H[h]，记录当前处于转换后的树的第几层，当h增加就接上H[h-1]+1，h减少直接退回，不用额外对H操作

代码

```
1 s = input()
2 h = a = b = 0
3 H = [0]*len(s)
4 for s in S:
5     if s == 'd':
6         h += 1
7         H[h] = H[h-1]+1
8         a = max(a, h)
9         b = max(b, H[h])
```

```

10         else:
11             h -= 1
12             H[h] += 1
13     print('%d => %d' % (a, b))
14

```

代码运行截图

#44655260提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

S = input()
h = a = b = 0
H = [0]*len(S)
for s in S:
    if s == 'd':
        h += 1
        H[h] = H[h-1]+1
        a = max(a, h)
        b = max(b, H[h])
    else:
        h -= 1
        H[h] += 1
print('%d => %d' % (a, b))

```

基本信息

#: 44655260

题目: 04081

提交人: 23n2300011505(12号娱乐选手)

内存: 3632kB

时间: 29ms

语言: Python3

提交时间: 2024-04-14 18:57:29

## 08581: 扩展二叉树

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

思路:

记录当前节点的高度，初始为2，当该位上是字母就高度加1，否则减1，当高度为1时，说明前序的左子树已经遍历完了，把前序的“中左右”换成中序的“左中右”或后序的“左右中”再拼起来，就是要输出的结果

代码

```

1 def F(s, k):
2     if len(s) == 1:
3         return ''
4     i, c = 1, 2
5     while c-1:
6         c -= 2*int(s[i] == '.')-1
7         i += 1
8     l = F(s[1:i], k)
9     r = F(s[i:], k)
10    if k:
11        return l+r+s[0]
12    else:
13        return l+s[0]+r
14

```

```
15
16 s = input()
17 print(F(s, 0))
18 print(F(s, 1))
19
```

代码运行截图

#44656013提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
def F(s, k):
    if len(s) == 1:
        return ''
    i, c = 1, 2
    while c-1:
        c -= 2*int(s[i] == '.')-1
        i += 1
    l = F(s[1:i], k)
    r = F(s[i:], k)
    if k:
        return l+r+s[0]
    else:
        return l+s[0]+r

s = input()
print(F(s, 0))
print(F(s, 1))
```

基本信息

#: 44656013  
题目: 08581  
提交人: 23n2300011505(12号娱乐选手)  
内存: 3632kB  
时间: 26ms  
语言: Python3  
提交时间: 2024-04-14 19:24:04

## 22067: 快速堆猪

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

思路:

用辅助栈，每次push时，在辅助栈中加入当前最轻的猪的体重，pop时也同步pop，这样栈顶始终是当前猪堆中最轻的体重，查询时直接输出即可

代码

```
1 l, s = [], []
2 while True:
3     try:
4         s = input()
5     except EOFError:
6         break
7     if s == 'min':
8         if s:
9             print(l[-1])
10    elif s == 'pop':
11        if s:
```

```

12         S.pop()
13         l.pop()
14     else:
15         n = int(s.split()[1])
16         S.append(n)
17         if l:
18             n = min(l[-1], n)
19         l.append(n)
20

```

代码运行截图

#44084790提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

l, S = [], []
while True:
    try:
        s = input()
    except EOFError:
        break
    if s == 'min':
        if S:
            print(l[-1])
    elif s == 'pop':
        if S:
            S.pop()
            l.pop()
    else:
        n = int(s.split()[1])
        S.append(n)
        if l:
            n = min(l[-1], n)
        l.append(n)

```

基本信息

#: 44084790

题目: 22067

提交人: 23n2300011505(12号娱乐选手)

内存: 5940kB

时间: 302ms

语言: Python3

提交时间: 2024-03-05 23:46:32

## 04123: 马走日

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

思路:

用dfs暴力搜索，能走的就试试，最后输出成功遍历的次数即可

代码

```

1 dx = [1, 1, 2, 2, -1, -1, -2, -2]
2 dy = [2, -2, 1, -1, 2, -2, 1, -1]
3
4
5 def dfs(n, m, x, y, gone):
6     if len(gone) == n*m:
7         return 1

```

```

8     a = 0
9     for i in range(8):
10         nx, ny = x+dx[i], y+dy[i]
11         idx = nx+ny*n
12         if 0 <= nx < n and 0 <= ny < m and idx not in gone:
13             gone.add(idx)
14             a += dfs(n, m, nx, ny, gone)
15             gone.remove(idx)
16     return a
17
18
19 for _ in range(int(input())):
20     n, m, x, y = map(int, input().split())
21     print(dfs(n, m, x, y, set([x+y*n])))
22

```

代码运行截图

#44656655提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

dx = [1, 1, 2, 2, -1, -1, -2, -2]
dy = [2, -2, 1, -1, 2, -2, 1, -1]

def dfs(n, m, x, y, gone):
    if len(gone) == n*m:
        return 1
    a = 0
    for i in range(8):
        nx, ny = x+dx[i], y+dy[i]
        idx = nx+ny*n
        if 0 <= nx < n and 0 <= ny < m and idx not in gone:
            gone.add(idx)
            a += dfs(n, m, nx, ny, gone)
            gone.remove(idx)
    return a

for _ in range(int(input())):
    n, m, x, y = map(int, input().split())
    print(dfs(n, m, x, y, set([x+y*n])))

```

基本信息

#: 44656655

题目: 04123

提交人: 23n2300011505(12号娱乐选手)

内存: 3680kB

时间: 4141ms

语言: Python3

提交时间: 2024-04-14 19:47:52

## 28046: 词梯

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

思路:

没有任何优化就会超时，一开始以为是bfs导致的超时，改了各种方法，甚至自己手搓了一个双端bfs，还是超时

后来意识到是建立连接的时候，两两都对比太慢了，于是使用桶的思想，把三个位置字母都相同的放到同一个桶里，最后再内部建立连接

代码

```
1  from collections import defaultdict as D
2  n = int(input())
3  l = [input() for _ in range(n)]
4  b, g, p = D(set), D(set), {}
5  for s in l:
6      b['_'+s[1:]].add(s)
7      b[s[0]+'_'+s[2:]].add(s)
8      b[s[:2]+'_'+s[3]].add(s)
9      b[s[:3]+'_'].add(s)
10 for a in b.values():
11     for x in a:
12         for y in a:
13             if x != y:
14                 g[x].add(y)
15 a, b = input().split()
16 if a == b:
17     print(a)
18     exit()
19 l = [a]
20 gone = set([a])
21 s, e = 0, 1
22 while s != e:
23     for i in range(s, e):
24         for a in g[l[i]]:
25             if a not in gone:
26                 p[a] = l[i]
27                 gone.add(a)
28                 l.append(a)
29             if a == b:
30                 ans = [b]
31                 while True:
32                     b = p[b]
33                     ans.append(b)
34                     if b == l[0]:
35                         break
36                 print(*ans[::-1])
37                 exit()
38     s, e = e, len(l)
39 print('NO')
```

状态: **Accepted**

源代码

```
from collections import defaultdict as D
n = int(input())
l = [input() for _ in range(n)]
b, g, p = D(set), D(set), {}
for s in l:
    b['_'+s[1:]].add(s)
    b[s[0]+'_'+s[2:]].add(s)
    b[s[:2]+'_'+s[3]].add(s)
    b[s[:3]+'_'+s[4]].add(s)
for a in b.values():
    for x in a:
        for y in a:
            if x != y:
                g[x].add(y)
a, b = input().split()
if a == b:
    print(a)
    exit()
l = [a]
gone = set([a])
s, e = 0, 1
while s != e:
    for i in range(s, e):
        for a in g[l[i]]:
            if a not in gone:
                p[a] = l[i]
                gone.add(a)
                l.append(a)
            if a == b:
                ans = [b]
                while True:
                    b = p[b]
                    ans.append(b)
                    if b == l[0]:
                        break
                print(*ans[::-1])
                exit()
    s, e = e, len(l)
print('NO')
```

基本信息

#: 44660278  
题目: 28046  
提交人: 23n2300011505(12号娱乐选手)  
内存: 10292kB  
时间: 61ms  
语言: Python3  
提交时间: 2024-04-14 23:33:32

## 28050: 骑士周游

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

思路:

一开始用传统的dfs, 卡了很久, 甚至尝试了打表, 都没成功 (不优化实在太慢了)

查了一下, 有一步优化是优先走下一步能走的格子少的格子, 虽然没想明白为什么, 但是加上之后速度发生了质的变化🤖

代码

```
1 dx = [1, 1, 2, 2, -1, -1, -2, -2]
2 dy = [2, -2, 1, -1, 2, -2, 1, -1]
3 n = int(input())
4 x, y = map(int, input().split())
```

```

5  g = [set() for _ in range(n*n)]
6  for i in range(n*n):
7      for k in range(8):
8          if 0 <= i//n+dx[k] < n and 0 <= i % n + dy[k] < n:
9              g[i].add(i+dx[k]*n+dy[k])
10
11  gone = [False]*(n*n)
12
13
14  def dfs(x, gone, k):
15      if not k:
16          print('success')
17          exit()
18      l = []
19      for a in g[x]:
20          if not gone[a]:
21              c = 0
22              for b in g[a]:
23                  if not gone[b]:
24                      c += 1
25              l.append((a, c))
26      l.sort(key=lambda x: x[1])
27      for a, _ in l:
28          gone[a] = True
29          dfs(a, gone, k-1)
30          gone[a] = False
31
32
33  gone[x*n+y] = True
34  dfs(x*n+y, gone, n*n-1)
35  print('fail')
36

```

代码运行截图



状态: **Accepted**

源代码

```
dx = [1, 1, 2, 2, -1, -1, -2, -2]
dy = [2, -2, 1, -1, 2, -2, 1, -1]
n = int(input())
x, y = map(int, input().split())
g = [set() for _ in range(n*n)]
for i in range(n*n):
    for k in range(8):
        if 0 <= i//n+dx[k] < n and 0 <= i % n + dy[k] < n:
            g[i].add(i+dx[k]*n+dy[k])

gone = [False]*(n*n)

def dfs(x, gone, k):
    if not k:
        print('success')
        exit()
    l = []
    for a in g[x]:
        if not gone[a]:
            c = 0
            for b in g[a]:
                if not gone[b]:
                    c += 1
            l.append((a, c))
    l.sort(key=lambda x: x[1])
    for a, _ in l:
        gone[a] = True
        dfs(a, gone, k-1)
        gone[a] = False

gone[x*n+y] = True
dfs(x*n+y, gone, n*n-1)
print('fail')
```

基本信息

#: 44660111  
题目: 28050  
提交人: 23n2300011505(12号娱乐选手)  
内存: 4104kB  
时间: 27ms  
语言: Python3  
提交时间: 2024-04-14 23:18:23

状态: **Accepted**

源代码

```
dx = [1, 1, 2, 2, -1, -1, -2, -2]
dy = [2, -2, 1, -1, 2, -2, 1, -1]
n = int(input())
x, y = map(int, input().split())
g = [set() for _ in range(n*n)]
for i in range(n*n):
    for k in range(8):
        if 0 <= i//n+dx[k] < n and 0 <= i % n + dy[k] < n:
            g[i].add(i+dx[k]*n+dy[k])

gone = [False]*(n*n)

def dfs(x, gone, k):
    if not k:
        print('success')
        exit()
    l = []
    for a in g[x]:
        if not gone[a]:
            c = 0
            for b in g[a]:
                if not gone[b]:
                    c += 1
            l.append((a, c))
    l.sort(key=lambda x: x[1])
    for a, _ in l:
        gone[a] = True
        dfs(a, gone, k-1)
        gone[a] = False

gone[x*n+y] = True
dfs(x*n+y, gone, n*n-1)
print('fail')
```

基本信息

#: 44660111  
题目: 28050  
提交人: 23n2300011505(12号娱乐选手)  
内存: 4104kB  
时间: 27ms  
语言: Python3  
提交时间: 2024-04-14 23:18:23

## 2. 学习总结和收获

被最后两题的优化卡了很久，可见我已经写惯了最常规的算法，优化思想已经生疏了，还需要多练习