

Assignment #A: 图论：算法，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Compiled by ==陈亚偲 工学院==

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

==（请改为同学的操作系统、编程环境等）==

操作系统：Windows

Python编程环境：Spyder IDE 5.2.2

1. 题目

20743: 整人的提词本

<http://cs101.openjudge.cn/practice/20743/>

思路：

类似双指针的操作，扫过去即可

代码

```
#
a=list(input())
index=0
while '(' in a:
    b=0
    c=0
    for i in range(len(a)):
        if a[i]=='(':
            b=i
        if a[i]==')':
            c=i
        break
    k=a[b+1:c]
    k.reverse()
```

```
a=a[:b]+k+a[c+1:]
print(''.join(a))
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
a=list(input())
index=0
while '(' in a:
    b=0
    c=0
    for i in range(len(a)):
        if a[i]=='(':
            b+=1
        if a[i]==')':
            c+=1
            break
    k=a[b+1:c]
    k.reverse()
    a=a[:b]+k+a[c+1:]
print(''.join(a))
```

基本信息

#: 44300402
题目: 20743
提交人: 23n2300011106(boza)
内存: 3608kB
时间: 23ms
语言: Python3
提交时间: 2024-03-19 17:10:59

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

02255: 重建二叉树

<http://cs101.openjudge.cn/practice/02255/>

思路:

和 22158:根据二叉树前中序序列建树 的代码几乎一模一样

```
#
while True:
    try:
        qwert=input().split()
        b=list(qwert[0])
        b.reverse()
        a=qwert[1]
        #
        k=len(b)
        u=[]
        class t:
            def __init__(self,name):
                self.n=name
                self.r=None
                self.l=None
        c=[t(i) for i in b]
        def d(m,p):
            temp=p.split(m)
            ll=list(temp[0])
            rr=list(temp[1])
            if ll:
                lm=max([b.index(i) for i in ll])
                c[b.index(m)].l=c[lm]
            if rr:
```

```

        rm=max([b.index(i) for i in rr])
        c[b.index(m)].r=c[rm]
    return
def bb(m,p):
    d(m,p)
    temp=p.split(m)
    if c[b.index(m)].r:
        bb(c[b.index(m)].r.n,temp[1])
    if c[b.index(m)].l:
        bb(c[b.index(m)].l.n,temp[0])
    return
def up(m):
    if m!=None:
        up(m.l)
        up(m.r)
        u.append(m.n)
    return
bb(c[-1].n,a)
up(c[-1])
print(''.join(u))
except:
    break

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```

while True:
    try:
        qwert=input().split()
        b=list(qwert[0])
        b.reverse()
        a=qwert[1]
        #
        k=len(b)
        u=[]
        class t:
            def __init__(self,name):
                self.n=name
                self.r=None
                self.l=None
        c=[t(i) for i in b]
        def d(m,p):
            temp=p.split(m)
            ll=list(temp[0])
            rr=list(temp[1])
            if ll:
                lm=max([b.index(i) for i in ll])
                c[b.index(m)].l=c[lm]
            if rr:
                rm=max([b.index(i) for i in rr])
                c[b.index(m)].r=c[rm]
            return
        def bb(m,p):
            d(m,p)
            temp=p.split(m)
            if c[b.index(m)].r:
                bb(c[b.index(m)].r.n,temp[1])
            if c[b.index(m)].l:
                bb(c[b.index(m)].l.n,temp[0])
            return
        def up(m):

```

基本信息

#: 44838316
 题目: 02255
 提交人: 23n2300011106(boza)
 内存: 3704kB
 时间: 19ms
 语言: Python3
 提交时间: 2024-04-30 22:06:34

01426: Find The Multiple

<http://cs101.openjudge.cn/practice/01426/>

要求用bfs实现

思路:

完全没有思路, 问的chat

chat的思路是直接一个一个试 (之前以为这个一定tle, 结果过了)

代码

```
#
def find_multiple(n):
    if n == 1:
        return "1"

    visited = [False] * n
    queue = [(1 % n, "1")]

    while queue:
        remainder, num = queue.pop(0)

        if remainder == 0:
            return num

        for digit in "01":
            next_remainder = (remainder * 10 + int(digit)) % n
            if not visited[next_remainder]:
                visited[next_remainder] = True
                queue.append((next_remainder, num + digit))

    while True:
        n = int(input())
        if n == 0:
            break
        print(find_multiple(n))
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
def find_multiple(n):
    if n == 1:
        return "1"

    visited = [False] * n
    queue = [(1 % n, "1")]

    while queue:
        remainder, num = queue.pop(0)

        if remainder == 0:
            return num

        for digit in "01":
            next_remainder = (remainder * 10 + int(digit)) % n
            if not visited[next_remainder]:
                visited[next_remainder] = True
                queue.append((next_remainder, num + digit))

    while True:
        n = int(input())
        if n == 0:
            break
        print(find_multiple(n))
```

基本信息

#: 44838705
题目: 01426
提交人: 23n2300011106(boza)
内存: 3572kB
时间: 41ms
语言: Python3
提交时间: 2024-04-30 23:43:29

04115: 鸣人和佐助

bfs, <http://cs101.openjudge.cn/practice/04115/>

思路:

不知道怎么处理墙

copy的题解代码

相当于对于每次变化重建一个广度优先搜索 (?)

代码

```
#
from collections import deque
class Node:
    def __init__(self, x, y, tools, steps):
        self.x = x
        self.y = y
        self.tools = tools
        self.steps = steps
M, N, T = map(int, input().split())
maze = [list(input()) for _ in range(M)]
visit = [[[0]*(T+1) for _ in range(N)] for _ in range(M)]
directions = [[-1, 0], [1, 0], [0, -1], [0, 1]]
start = end = None
flag = 0
for i in range(M):
    for j in range(N):
        if maze[i][j] == '@':
            start = Node(i, j, T, 0)
            visit[i][j][T] = 1
        if maze[i][j] == '+':
            end = (i, j)
```

```

        maze[i][j] = '*'

queue = deque([start])
while queue:
    node = queue.popleft()
    if (node.x, node.y) == end:
        print(node.steps)
        flag = 1
        break
    for direction in directions:
        nx, ny = node.x+direction[0], node.y+direction[1]
        if 0 <= nx < M and 0 <= ny < N:
            if maze[nx][ny] == '*':
                if not visit[nx][ny][node.tools]:
                    queue.append(Node(nx, ny, node.tools, node.steps+1))
                    visit[nx][ny][node.tools] = 1
            elif maze[nx][ny] == '#':
                if node.tools > 0 and not visit[nx][ny][node.tools-1]:
                    queue.append(Node(nx, ny, node.tools-1, node.steps+1))
                    visit[nx][ny][node.tools-1] = 1

if not flag:
    print("-1")

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```

from collections import deque
class Node:
    def __init__(self, x, y, tools, steps):
        self.x = x
        self.y = y
        self.tools = tools
        self.steps = steps
M, N, T = map(int, input().split())
maze = [list(input()) for _ in range(M)]
visit = [[[0]*(T+1) for _ in range(N)] for _ in range(M)]
directions = [[-1, 0], [1, 0], [0, -1], [0, 1]]
start = end = None
flag = 0
for i in range(M):
    for j in range(N):
        if maze[i][j] == '@':
            start = Node(i, j, T, 0)
            visit[i][j][T] = 1
        if maze[i][j] == '+':
            end = (i, j)
            maze[i][j] = '*'

queue = deque([start])
while queue:
    node = queue.popleft()
    if (node.x, node.y) == end:
        print(node.steps)
        flag = 1
        break
    for direction in directions:
        nx, ny = node.x+direction[0], node.y+direction[1]
        if 0 <= nx < M and 0 <= ny < N:
            if maze[nx][ny] == '*':
                if not visit[nx][ny][node.tools]:
                    queue.append(Node(nx, ny, node.tools, node.steps+1))

```

基本信息

#: 44838693
 题目: 04115
 提交人: 23n2300011106(boza)
 内存: 7236kB
 时间: 116ms
 语言: Python3
 提交时间: 2024-04-30 23:39:52

20106: 走山路

Dijkstra, <http://cs101.openjudge.cn/practice/20106/>

思路:

传统bfs会wa

代码

```
#
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

05442: 兔子与星空

Prim, <http://cs101.openjudge.cn/practice/05442/>

思路:

代码

```
#
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

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。==

迪杰斯特拉算法欠缺较大

合适的时候可以建立一个三位数组来跑有特殊限制的bfs