Assignment #A: 图论：算法，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Complied by 同学的姓名、院系

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：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

20743: 整人的提词本

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

思路：

栈，非右括号入栈，遇到右括号开始对栈循环弹出直到为左括号，删除左括号，把之前弹出的重新放到左边，直到不再输入

代码

```
s=input()
ipt_stack=[x for x in s]
stack=[]
while ipt_stack:
    p=ipt_stack.pop(0)
    if p!=')':
        stack.append(p)
    else:
        pre = []
        while stack[-1] != '(':
```

```
            pre.append(stack.pop())
        stack.pop(-1)
        if ipt_stack:
            pre.extend(ipt_stack)
            ipt_stack=pre
        else:
            stack.extend(pre)
#     print('stack:',stack)
#     print('ipt_stack:',ipt_stack)
#     print('------------------------')
print(''.join(stack))
```

代码运行截图 （至少包含有"Accepted"）

02255: 重建二叉树

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

思路:

前中序转后序，之前写过

代码

```
class Node:
    def __init__(self,v):
        self.value=v
        self.left=None
        self.right=None

    def post_order(self):
        ans=[]
        if self.left:
            ans.extend(self.left.post_order())
        if self.right:
            ans.extend(self.right.post_order())
        ans.append(self.value)
        return ans

def BuildTree(prelst,inlst):
    if len(prelst)==1:
        return Node(prelst[0])
    elif not prelst:
        return None
    else:
        r=prelst[0]
        node=Node(r)
        idx=inlst.index(r)
        lp=prelst[1:idx+1]
        li=inlst[:idx]
        rp=prelst[idx+1:]
        ri=inlst[idx+1:]
        node.left=BuildTree(lp,li)
        node.right=BuildTree(rp,ri)
        return node

ans=[]
while True:
    try:
        prestr,instr=map(str,input().split())
        prelst=[x for x in prestr]
        inlst=[y for y in instr]
        tree=BuildTree(prelst,inlst)
        ans.append(tree.post_order())
    except EOFError:
        break

for x in ans:
```

```
    print(''.join(x))
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
class Node:
    def __init__(self,v):
        self.value=v
        self.left=None
        self.right=None

    def post_order(self):
        ans=[]
        if self.left:
            ans.extend(self.left.post_order())
        if self.right:
            ans.extend(self.right.post_order())
        ans.append(self.value)
        return ans

def BuildTree(prelst,inlst):
    if len(prelst)==1:
        return Node(prelst[0])
    elif not prelst:
        return None
    else:
        r=prelst[0]
        node=Node(r)
        idx=inlst.index(r)
        lp=prelst[1:idx+1]
        li=inlst[:idx]
        rp=prelst[idx+1:]
        ri=inlst[idx+1:]
        node.left=BuildTree(lp,li)
        node.right=BuildTree(rp,ri)
        return node

ans=[]
while True:
    try:
        prestr,instr=map(str,input().split())
        prelst=[x for x in prestr]
        inlst=[y for y in instr]
        tree=BuildTree(prelst,inlst)
        ans.append(tree.post_order())
    except EOFError:
        break
```

基本信息

#: 44831
题目: 02255
提交人: zxk
内存: 3616k
时间: 21ms
语言: Pytho
提交时间: 2024-

01426: Find The Multiple

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

要求用bfs实现

思路：

感谢要求合理即可，其实不太分得开这里bfs和dfs的区别啊？我理解的bfs是 0，1，10，11，100，111 这种吧，感觉实现起来也会容易；由于懒得想怎么实现只有1，0的bfs，所以干脆用二进制数（，在十进制里加一，其实就是索引的指标变了一个hhh，略微实验了一下这个方法还真行欸！

代码

```
def show(i):
    return int(bin(i)[2:])
def bfs(n):
    i=1
    while show(i)%n!=0:
        i+=1
    return show(i)

ans=[]
while True:
    n=int(input())
    if n==0:
        break
    else:
        ans.append(bfs(n))

for x in ans:
    print(x)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

CS101 / 题库

题目 排名 状态 提问

#44831626提交状态 查看

状态: Accepted

源代码

```
def show(i):
    return int(bin(i)[2:])
def bfs(n):
    i=1
    while show(i)%n!=0:
        i+=1
    return show(i)

ans=[]
while True:
    n=int(input())
    if n==0:
        break
    else:
        ans.append(bfs(n))

for x in ans:
    print(x)
```

基本信息

#: 448316
题目: 01426
提交人: zxk
内存: 3520kB
时间: 805ms
语言: Python3
提交时间: 2024-0

04115: 鸣人和佐助

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

思路:

代码

```

```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

20106: 走山路

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

思路:

抄的答案

代码

```
from heapq import heappop, heappush

def bfs(x1, y1):
    vertex = [(0, x1, y1)]
    visited = [[False] * n for _ in range(m)]
    while vertex:
        t, x, y = heappop(vertex)
        if visited[x][y]:
            continue
        visited[x][y]=True
        if x == x2 and y == y2:
            return t
        for dx, dy in directions:
            nx, ny = x+dx, y+dy
            if 0 <= nx < m and 0 <= ny < n and maplst[nx][ny] != '#' and not
visited[nx][ny]:
                nt = t+abs(int(maplst[nx][ny])-int(maplst[x][y]))
                heappush(vertex, (nt, nx, ny))
    return 'NO'


m, n, p = map(int, input().split())
maplst = [list(input().split()) for _ in range(m)]
directions = [(1, 0), (-1, 0), (0, 1), (0, -1)]
for _ in range(p):
    x1, y1, x2, y2 = map(int, input().split())
    if maplst[x1][y1] == '#' or maplst[x2][y2] == '#':
        print('NO')
        continue
    print(bfs(x1, y1))
```

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。

1.整人的提词本：好久没有做栈的题，已经很生疏了，花了不少时间才想出来办法；对于左右括号的题感觉重点是处理对于先输入的情况的记录和右括号出现后如何处理使得这个括号的影响消除，以便于循环

2.重建二叉树：没啥新东西，原题，自己编了一遍比之前顺多了

3.Find The Multiple：最简单的一集，得亏是没卡怎么找出来的是合理的，靠二进制当指标循环一下就找到了（甚至没递归

5.走山路：自己写了半天dijkstra结果只是bfs就可以？？区别是bfs不是按照相邻节点的顺序，而是按照最短距离的顺序，好妙的做法……我自己做dijkstra的方法会超时，感觉是因为找下一个处理的节点的时候相当于对整个表的距离进行了排序，耗时很大，但是后来想给改heap发现代码语言就很乱，而且卡在了如何排距离的同时保留节点坐标的信息上，没想到heapq竟然可以对三元数组排序吗？

五一期间事情比较多，上周又考试，课掌握的不好，ddl已经到了只好先交上四个题，剩下鸣人佐助和星空五一快结束的时候补