Assignment #A: 图论：遍历，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Complied by 郑铭毅 数学科学学院

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 typorahttps://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 11

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

20743: 整人的提词本

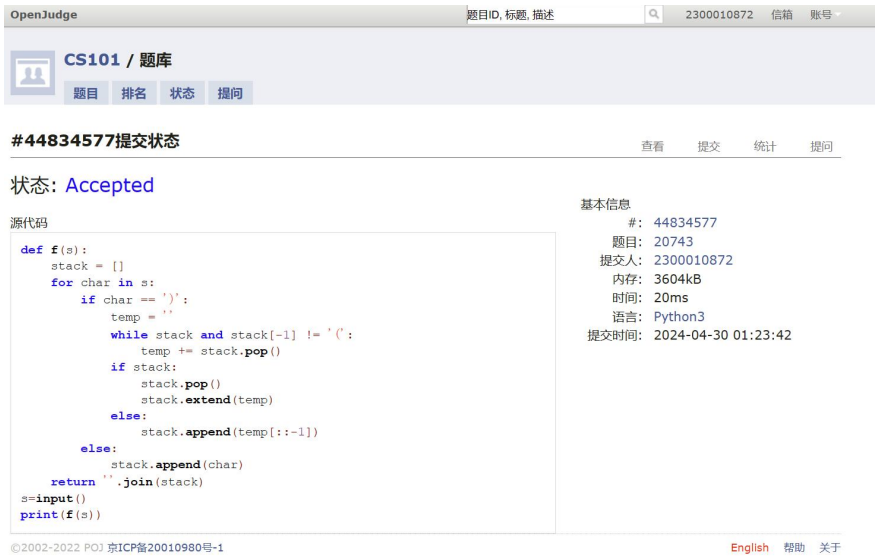
http://cs101.openjudge.cn/practice/20743/

思路:

代码

```
def f(s):
    stack = []
    for char in s:
        if char == ')':
            temp = ''
            while stack and stack[-1] != '(':
                temp += stack.pop()
            if stack:
                stack.pop()
                stack.extend(temp)
            else:
                stack.append(temp[::-1])
        else:
            stack.append(char)
    return ''.join(stack)
s=input()
print(f(s))
```

代码运行截图



02255: 重建二叉树

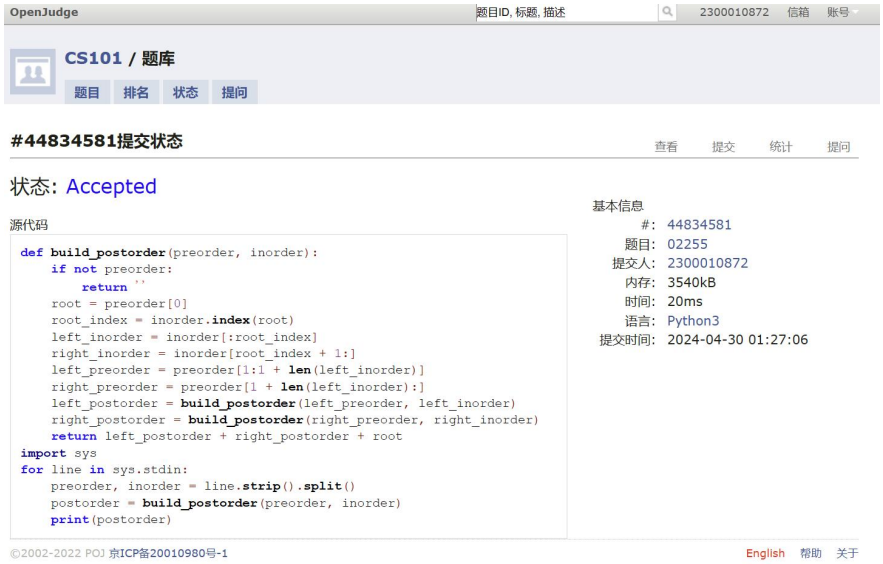
http://cs101.openjudge.cn/practice/02255/

思路:

代码

```
def build_postorder(preorder, inorder):
    if not preorder:
        return ''
    root = preorder[0]
    root_index = inorder.index(root)
    left_inorder = inorder[:root_index]
    right_inorder = inorder[root_index + 1:]
    left_preorder = preorder[1:1 + len(left_inorder)]
    right_preorder = preorder[1 + len(left_inorder):]
    left_postorder = build_postorder(left_preorder, left_inorder)
    right_postorder = build_postorder(right_preorder, right_inorder)
    return left_postorder + right_postorder + root
import sys
for line in sys.stdin:
    preorder, inorder = line.strip().split()
    postorder = build_postorder(preorder, inorder)
    print(postorder)
```

代码运行截图



01426: Find The Multiple

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

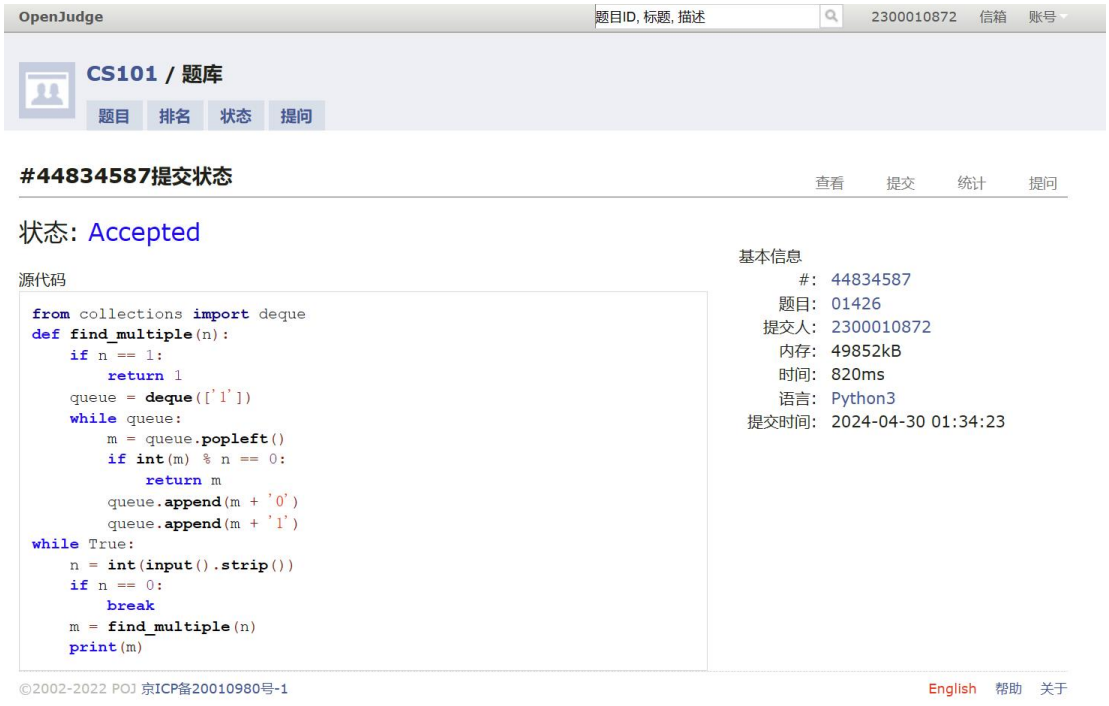
要求用bfs实现

思路:

代码

```
from collections import deque
def find_multiple(n):
    if n == 1:
        return 1
    queue = deque(['1'])
    while queue:
        m = queue.popleft()
        if int(m) % n == 0:
            return m
        queue.append(m + '0')
        queue.append(m + '1')
while True:
    n = int(input().strip())
    if n == 0:
        break
    m = find_multiple(n)
    print(m)
```

代码运行截图

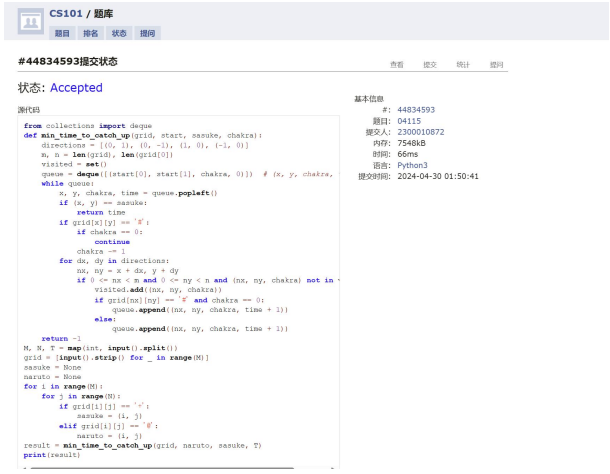


04115: 鸣人和佐助

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

思路:

```
from collections import deque
def min_time_to_catch_up(grid, start, sasuke, chakra):
    directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]
    m, n = len(grid), len(grid[0])
    visited = set()
    queue = deque([(start[0], start[1], chakra, 0)])  # (x, y, chakra, time)
    while queue:
        x, y, chakra, time = queue.popleft()
        if (x, y) == sasuke:
            return time
        if grid[x][y] == '#':
            if chakra == 0:
                continue
            chakra -= 1
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < m and 0 <= ny < n and (nx, ny, chakra) not in visited:
                visited.add((nx, ny, chakra))
                if grid[nx][ny] == '#' and chakra == 0:
                    queue.append((nx, ny, chakra, time + 1))
                else:
                    queue.append((nx, ny, chakra, time + 1))
    return -1
M, N, T = map(int, input().split())
grid = [input().strip() for _ in range(M)]
sasuke = None
naruto = None
for i in range(M):
    for j in range(N):
        if grid[i][j] == '+':
            sasuke = (i, j)
        elif grid[i][j] == '@':
            naruto = (i, j)
result = min_time_to_catch_up(grid, naruto, sasuke, T)
print(result)
```

代码运行截图 

20106: 走山路

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

思路：

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

本周作业也挺有难度，花费的时间也比较多，假期要多花时间思考如何做题。