Assignment8

姓名：尹廖丹华

学号：2000092113

院系：心理与认知科学学院

一、 作业题目

1、04082:树的镜面映射（2h50min）

```
from collections import deque

class TreeNode:
    def __init__(self, x):
        self.x = x
        self.children = []

def create_node():
    return TreeNode('')

def build_tree(tempList, index):
    node = create_node()
    node.x = tempList[index][0]
    if tempList[index][1] == '0' and node.x != '$':
        index += 1
        child, index = build_tree(tempList, index)
        node.children.append(child)
        index += 1
        child, index = build_tree(tempList, index)
        node.children.append(child)
    return node, index

def print_tree(p):
    Q = deque()
    s = deque()

    while p is not None:
        if p.x != '$':
            s.append(p)
        p = p.children[1] if len(p.children) > 1 else None

    while s:
        Q.append(s.pop())

    while Q:
        p = Q.popleft()
        print(p.x, end=' ')

        if p.children:
            p = p.children[0]
            while p is not None:
                if p.x != '$':
```

```
                s.append(p)
            p = p.children[1] if len(p.children) > 1 else None

        while s:
            Q.append(s.pop())

n = int(input())
tempList = input().split(' ')

root, _ = build_tree(tempList, 0)

print_tree(root)
```

2、04089:电话号码（40min）

```
def is_consistent(numbers):
    numbers.sort()
    for i in range(len(numbers) - 1):
        if numbers[i+1].startswith(numbers[i]):
            return "NO"
    return "YES"

t = int(input())
for _ in range(t):
    n = int(input())
    phone_numbers = [input() for _ in range(n)]
    print(is_consistent(phone_numbers))
```

3、18160:最大连通域面积(matrix,dfs)（2h30min）

```
def dfs(grid, i, j):
    if i < 0 or i >= len(grid) or j < 0 or j >= len(grid[0]) or
grid[i][j] != 'W':
        return 0
    grid[i][j] = '.'
    count = 1
    for dx in [-1, 0, 1]:
        for dy in [-1, 0, 1]:
            count += dfs(grid, i + dx, j + dy)
    return count

def max_connected_area(grid):
    max_area = 0
    for i in range(len(grid)):
        for j in range(len(grid[0])):
            if grid[i][j] == 'W':
                max_area = max(max_area, dfs(grid, i, j))
    return max_area

T = int(input())
for _ in range(T):
    N, M = map(int, input().split())
    grid = [list(input()) for _ in range(N)]
```

```
    print(max_connected_area(grid))
```

4、19943:图的拉普拉斯矩阵（3h）

```
def laplacian_matrix(n, edges):
    laplacian = [[0] * n for _ in range(n)]
    for i in range(n):
        degree = sum(1 for edge in edges if i in edge)
        laplacian[i][i] = degree

    for edge in edges:
        i, j = edge
        laplacian[i][j] -= 1
        laplacian[j][i] -= 1
    return laplacian

n, m = map(int, input().split())
edges = [tuple(map(int, input().split())) for _ in range(m)]

laplacian = laplacian_matrix(n, edges)
for row in laplacian:
    print(*row)
```

二、 学习的收获与感受

一开始都在自己慢慢摸索，但是发现耗时太长了，而且有些题目没有找到题解，幸好 gpt 写的代码能 ac，就看着学习了一下。图和树都好难，找时间还是得看看教材巩固一下基础。