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==

罗瑞哲 生命科学学院

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 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 10

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 reverse_parentheses(s):
    stack = []
    for char in s:
        if char == ')':
            temp = []
            while stack and stack[-1] != '(':
                temp.append(stack.pop())
            if stack:
                stack.pop()
            stack.extend(temp)
```

```
        else:
            stack.append(char)
    return ''.join(stack)
s = input().strip()
print(reverse_parentheses(s))
```

代码运行截图 ==（至少包含有"Accepted"）==

#44802255提交状态

状态: Accepted

源代码

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

02255: 重建二叉树

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

思路:

代码

```
#
class TreeNode:
    def __init__(self,val):
        self.val=val
        self.left=None
        self.right=None
def build_tree(preorder,inorder):
    if not preorder or not inorder:
        return None
    r=preorder[0]
    root=TreeNode(r)
```

```
    id=inorder.index(r)
    root.left=build_tree(preorder[1:1+id],inorder[:id])
    root.right=build_tree(preorder[1+id:],inorder[id+1:])
    return root
def postorder(root):
    if not root:
        return ""
    return postorder(root.left)+postorder(root.right)+root.val
while True:
    try:
        preorder,inorder=map(str,input().split())
        root=build_tree(preorder,inorder)
        ans=postorder(root)
        print(ans)
    except EOFError:
        break
```

代码运行截图 ==（至少包含有"Accepted"）==

#44802524提交状态

状态: Accepted

源代码

```
class TreeNode:
    def __init__(self,val):
        self.val=val
        self.left=None
        self.right=None
def build_tree(preorder,inorder):
    if not preorder or not inorder:
        return None
    r=preorder[0]
    root=TreeNode(r)
    id=inorder.index(r)
    root.left=build_tree(preorder[1:1+id],inorder[:id])
    root.right=build_tree(preorder[1+id:],inorder[id+1:])
    return root
def postorder(root):
    if not root:
        return ""
    return postorder(root.left)+postorder(root.right)+root.val
while True:
    try:
        preorder,inorder=map(str,input().split())
        root=build_tree(preorder,inorder)
        ans=postorder(root)
        print(ans)
    except EOFError:
        break
```

01426: Find The Multiple

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

要求用bfs实现

思路：

代码

```
#
def bfs(n):
    l = [0]
    s, e = 0, 1
    while s != e:
        for i in range(s, e):
            for j in (0, 1):
                x = l[i]*10+j
                if x:
                    if x % n:
                        l.append(x)
                    else:
                        return str(x)
        s, e = e, len(l)
    return ''
while (n := int(input())):
    c = 0
    while (n+1) % 2:
        n //= 2
        c += 1
    print(bfs(n)+'0'*c)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44802981提交状态

状态: Accepted

源代码

```
def bfs(n):
    l = [0]
    s, e = 0, 1
    while s != e:
        for i in range(s, e):
            for j in (0, 1):
                x = l[i]*10+j
                if x:
                    if x % n:
                        l.append(x)
                    else:
                        return str(x)
        s, e = e, len(l)
    return ''
while (n := int(input())):
    c = 0
    while (n+1) % 2:
        n //= 2
        c += 1
    print(bfs(n)+'0'*c)
```

04115: 鸣人和佐助

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

思路:

代码

```
#
from collections import deque
M, N, T = map(int, input().split())
graph = [list(input()) for i in range(M)]
direc = [(0,1), (1,0), (-1,0), (0,-1)]
start, end = None, None
for i in range(M):
    for j in range(N):
        if graph[i][j] == '@':
            start = (i, j)
def bfs():
    q = deque([start + (T, 0)])
    visited = [[-1]*N for i in range(M)]
    visited[start[0]][start[1]] = T
    while q:
        x, y, t, time = q.popleft()
```

```
        time += 1
        for dx, dy in direc:
            if 0<=x+dx<M and 0<=y+dy<N:
                if (elem := graph[x+dx][y+dy]) == '*' and t > visited[x+dx]
[y+dy]:
                    visited[x+dx][y+dy] = t
                    q.append((x+dx, y+dy, t, time))
                elif elem == '#' and t > 0 and t-1 > visited[x+dx][y+dy]:
                    visited[x+dx][y+dy] = t-1
                    q.append((x+dx, y+dy, t-1, time))
                elif elem == '+':
                    return time
    return -1
print(bfs())
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44802999提交状态

状态: Accepted

源代码

```
from collections import deque
M, N, T = map(int, input().split())
graph = [list(input()) for i in range(M)]
direc = [(0,1), (1,0), (-1,0), (0,-1)]
start, end = None, None
for i in range(M):
    for j in range(N):
        if graph[i][j] == '@':
            start = (i, j)
def bfs():
    q = deque([start + (T, 0)])
    visited = [[-1]*N for i in range(M)]
    visited[start[0]][start[1]] = T
    while q:
        x, y, t, time = q.popleft()
        time += 1
        for dx, dy in direc:
            if 0<=x+dx<M and 0<=y+dy<N:
                if (elem := graph[x+dx][y+dy]) == '*' and t > visited[x+
                    visited[x+dx][y+dy] = t
                    q.append((x+dx, y+dy, t, time))
                elif elem == '#' and t > 0 and t-1 > visited[x+dx][y+dy]
                    visited[x+dx][y+dy] = t-1
                    q.append((x+dx, y+dy, t-1, time))
                elif elem == '+':
                    return time
    return -1
print(bfs())
```

20106: 走山路

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

思路:

代码

```
#
def bfs(x, y):
    directions = [(0, -1), (0, 1), (1, 0), (-1, 0)]
    queue = [(x, y)]
    distances = {(x, y): 0}
    while queue:
        current_x, current_y = queue.pop(0)
        for dx, dy in directions:
            new_x, new_y = current_x + dx, current_y + dy
            if 0 <= new_x < m and 0 <= new_y < n:
                if d[new_x][new_y] != '#':
                    new_distance = distances[(current_x, current_y)] +
abs(int(d[new_x][new_y]) - int(d[current_x][current_y]))
                    if (new_x, new_y) not in distances or new_distance <
distances[(new_x, new_y)]:
                        distances[(new_x, new_y)] = new_distance
                        queue.append((new_x, new_y))
    return distances
m, n, p = map(int, input().split())
d = []
for _ in range(m):
    row = input().split()
    d.append(row)
for _ in range(p):
    x1, y1, x2, y2 = map(int, input().split())
    if d[x1][y1] == '#' or d[x2][y2] == '#':
        print('NO')
        continue
    distances = bfs(x1, y1)
    if (x2, y2) in distances:
        print(distances[(x2, y2)])
    else:
        print('NO')
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44803016提交状态

状态: Accepted

源代码

```
def bfs(x, y):
    directions = [(0, -1), (0, 1), (1, 0), (-1, 0)]
    queue = [(x, y)]
    distances = {(x, y): 0}
    while queue:
        current_x, current_y = queue.pop(0)
        for dx, dy in directions:
            new_x, new_y = current_x + dx, current_y + dy
            if 0 <= new_x < m and 0 <= new_y < n:
                if d[new_x][new_y] != '#':
                    new_distance = distances[(current_x, current_y)] + a
                    if (new_x, new_y) not in distances or new_distance <
                        distances[(new_x, new_y)] = new_distance
                        queue.append((new_x, new_y))
    return distances
m, n, p = map(int, input().split())
d = []
for _ in range(m):
    row = input().split()
    d.append(row)
for _ in range(p):
    x1, y1, x2, y2 = map(int, input().split())
    if d[x1][y1] == '#' or d[x2][y2] == '#':
        print('NO')
        continue
    distances = bfs(x1, y1)
    if (x2, y2) in distances:
        print(distances[(x2, y2)])
    else:
        print('NO')
```

05442: 兔子与星空

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

思路:

代码

```
#
import heapq
def prim(graph, start):
    mst = []
    used = set([start])
```

```
    edges = [
        (cost, start, to)
        for to, cost in graph[start].items()
    ]
    heapq.heapify(edges)
    while edges:
        cost, frm, to = heapq.heappop(edges)
        if to not in used:
            used.add(to)
            mst.append((frm, to, cost))
            for to_next, cost2 in graph[to].items():
                if to_next not in used:
                    heapq.heappush(edges, (cost2, to, to_next))
    return mst
def solve():
    n = int(input())
    graph = {chr(i+65): {} for i in range(n)}
    for i in range(n-1):
        data = input().split()
        star = data[0]
        m = int(data[1])
        for j in range(m):
            to_star = data[2+j*2]
            cost = int(data[3+j*2])
            graph[star][to_star] = cost
            graph[to_star][star] = cost
    mst = prim(graph, 'A')
    print(sum(x[2] for x in mst))
solve()
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44803027提交状态

状态：Accepted

源代码

```
import heapq
def prim(graph, start):
    mst = []
    used = set([start])
    edges = [
        (cost, start, to)
        for to, cost in graph[start].items()
    ]
    heapq.heapify(edges)
    while edges:
        cost, frm, to = heapq.heappop(edges)
        if to not in used:
            used.add(to)
            mst.append((frm, to, cost))
            for to_next, cost2 in graph[to].items():
                if to_next not in used:
                    heapq.heappush(edges, (cost2, to, to_next))
    return mst
def solve():
    n = int(input())
    graph = {chr(i+65): {} for i in range(n)}
    for i in range(n-1):
        data = input().split()
        star = data[0]
        m = int(data[1])
        for j in range(m):
            to_star = data[2+j*2]
            cost = int(data[3+j*2])
            graph[star][to_star] = cost
            graph[to_star][star] = cost
    mst = prim(graph, 'A')
    print(sum(x[2] for x in mst))
solve()
```

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。==

我准备利用五一假期恶补数算，包括我不熟悉的各种数据结构和算法等等。