Assignment #A: 图论：遍历，树算及栈

2024 spring, Complied by 胡豪俊 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

操作系统：Windows11

Python编程环境：Visual Studio Code

1. 题目

20743: 整人的提词本

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

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))
##写起来还是比较直接的，复习了一下栈
```

02255: 重建二叉树

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

代码

```
def build_tree(preorder, inorder):
    if not preorder:
        return ''

    root = preorder[0]
    root_index = inorder.index(root)

    left_preorder = preorder[1:1 + root_index]
    right_preorder = preorder[1 + root_index:]

    left_inorder = inorder[:root_index]
    right_inorder = inorder[root_index + 1:]

    left_tree = build_tree(left_preorder, left_inorder)
    right_tree = build_tree(right_preorder, right_inorder)

    return left_tree + right_tree + root

while True:
    try:
        preorder, inorder = input().split()
        postorder = build_tree(preorder, inorder)
        print(postorder)
    except EOFError:
        break
##复习了一下树的知识，和前面比起来更加熟悉了
```

#44836742提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def build_tree(preorder, inorder):
    if not preorder:
        return ''

    root = preorder[0]
    root_index = inorder.index(root)

    left_preorder = preorder[1:1 + root_index]
    right_preorder = preorder[1 + root_index:]

    left_inorder = inorder[:root_index]
    right_inorder = inorder[root_index + 1:]

    left_tree = build_tree(left_preorder, left_inorder)
    right_tree = build_tree(right_preorder, right_inorder)

    return left_tree + right_tree + root

while True:
    try:
        preorder, inorder = input().split()
        postorder = build_tree(preorder, inorder)
        print(postorder)
    except EOFError:
        break
```

基本信息

#: 44836742
题目: 02255
提交人: 22n2200011800
内存: 3568kB
时间: 21ms
语言: Python3
提交时间: 2024-04-30 17:18:57

English 帮助 关于

01426: Find The Multiple

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

代码

```
from collections import deque

def find_multiple(n):
    q = deque()
    q.append((1 % n, "1"))
    visited = set([1 % n])

    while q:
        mod, num_str = q.popleft()
        if mod == 0:
            return num_str

        for digit in ["0", "1"]:
            new_num_str = num_str + digit
            new_mod = (mod * 10 + int(digit)) % n

            if new_mod not in visited:
                q.append((new_mod, new_num_str))
                visited.add(new_mod)

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

```
if __name__ == "__main__":
    main()
##学习了题解
```

#44836747提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque

def find_multiple(n):
    q = deque()
    q.append((1 % n, "1"))
    visited = set([1 % n])

    while q:
        mod, num_str = q.popleft()
        if mod == 0:
            return num_str

        for digit in ["0", "1"]:
            new_num_str = num_str + digit
            new_mod = (mod * 10 + int(digit)) % n

            if new_mod not in visited:
                q.append((new_mod, new_num_str))
                visited.add(new_mod)

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

if __name__ == "__main__":
    main()
```

基本信息

#: 44836747
题目: 01426
提交人: 22n2200011800
内存: 3596kB
时间: 44ms
语言: Python3
提交时间: 2024-04-30 17:20:24

English 帮助 关于

04115: 鸣人和佐助

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

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())
##用deque实现还是挺方便的，bfs和dfs的实现方式感觉都是多种多样，每种都要学会
##同学们的代码简洁而优美而且用了很多不同的方法
```

#44836800提交状态

查看 提交 统计 提问

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())
```

基本信息

#: 44836800
题目: 04115
提交人: 22n2200011800
内存: 4088kB
时间: 63ms
语言: Python3
提交时间: 2024-04-30 17:28:25

English 帮助 关于

20106: 走山路

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

代码

```
import heapq
m, n, p = map(int, input().split())
martix = [list(input().split())for i in range(m)]
dir = [(-1, 0), (1, 0), (0, 1), (0, -1)]
for _ in range(p):
    sx, sy, ex, ey = map(int, input().split())
    if martix[sx][sy] == "#" or martix[ex][ey] == "#":
        print("NO")
        continue
    vis, heap, ans = set(), [], []
    heapq.heappush(heap, (0, sx, sy))
```

```
    vis.add((sx, sy, -1))
    while heap:
        tire, x, y = heapq.heappop(heap)
        if x == ex and y == ey:
            ans.append(tire)
        for i in range(4):
            dx, dy = dir[i]
            x1, y1 = dx+x, dy+y
            if 0 <= x1 < m and 0 <= y1 < n and martix[x1][y1] != "#" and (x1, y1,
i) not in vis:
                t1 = tire+abs(int(martix[x][y])-int(martix[x1][y1]))
                heapq.heappush(heap, (t1, x1, y1))
                vis.add((x1, y1, i))
    print(min(ans) if ans else "NO")
##之前没写过，看群里大家的反应这题应该会让人印象深刻，果然
```

#44836806提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq
m, n, p = map(int, input().split())
martix = [list(input().split())for i in range(m)]
dir = [(-1, 0), (1, 0), (0, 1), (0, -1)]
for _ in range(p):
    sx, sy, ex, ey = map(int, input().split())
    if martix[sx][sy] == "#" or martix[ex][ey] == "#":
        print("NO")
        continue
    vis, heap, ans = set(), [], []
    heapq.heappush(heap, (0, sx, sy))
    vis.add((sx, sy, -1))
    while heap:
        tire, x, y = heapq.heappop(heap)
        if x == ex and y == ey:
            ans.append(tire)
        for i in range(4):
            dx, dy = dir[i]
            x1, y1 = dx+x, dy+y
            if 0 <= x1 < m and 0 <= y1 < n and martix[x1][y1] != "#" and
                t1 = tire+abs(int(martix[x][y])-int(martix[x1][y1]))
                heapq.heappush(heap, (t1, x1, y1))
                vis.add((x1, y1, i))
    print(min(ans) if ans else "NO")
```

基本信息

#: 44836806
题目: 20106
提交人: 22n2200011800
内存: 4684kB
时间: 1642ms
语言: Python3
提交时间: 2024-04-30 17:29:34

English 帮助 关于

05442: 兔子与星空

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

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()
##学习了题解
```

#44836811提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```python
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()
```

基本信息

#: 44836811
题目: 05442
提交人: 22n2200011800
内存: 3676kB
时间: 21ms
语言: Python3
提交时间: 2024-04-30 17:30:40

English 帮助 关于

2. 学习总结和收获

在外旅游中，明天回校开卷