

# Assignment #A: 图论：遍历，树算及栈

Updated 2018 GMT+8 Apr 21, 2024

2024 spring, Compiled by Xinjie Song, Phy

## 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：Windows 11 22H2

Python编程环境：PyCharm 2023.2 (Community Edition)

C/C++编程环境：g++ (x86\_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

## 1. 题目

### 20743: 整人的提词本

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

思路：递归

代码

```
s = input()

def rev(s):
    idx = cnt = 0
    ans = ''
    for i in range(len(s)):
        if s[i] == '(':
            cnt += 1
            if cnt == 1:
                idx = i
        elif s[i] == ')':
            cnt -= 1
            if cnt == 0:
```

```

        ans += rev(s[idx + 1: i])[::-1]

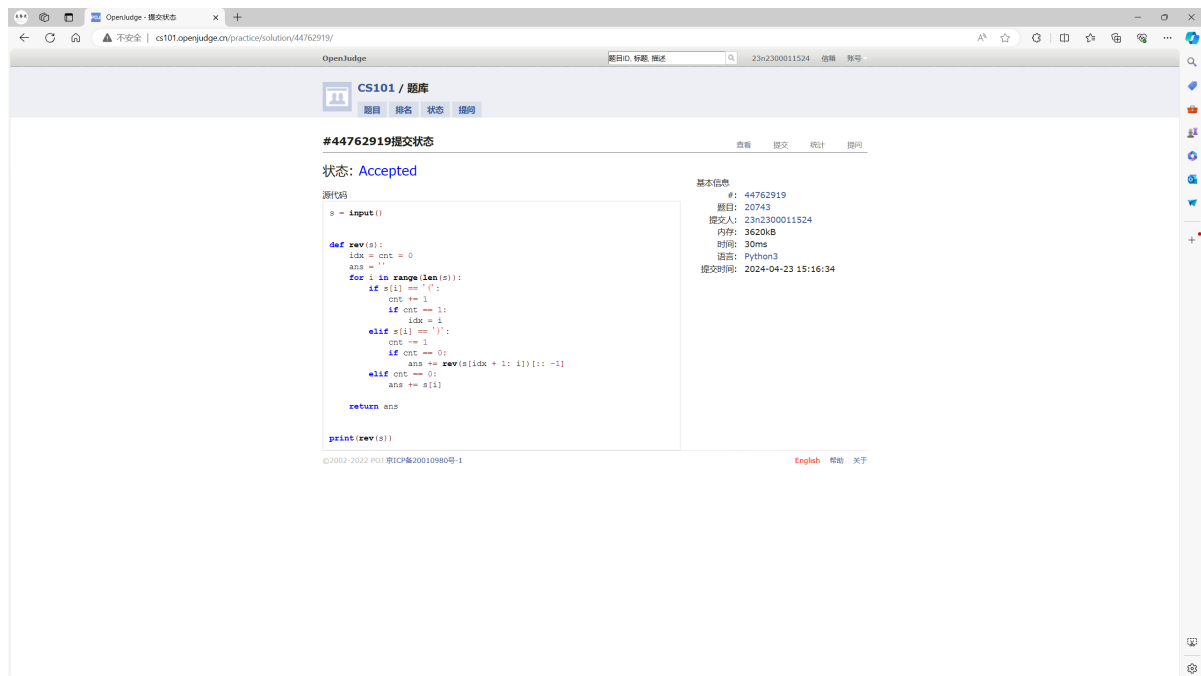
    elif cnt == 0:
        ans += s[i]

    return ans

print(rev(s))

```

代码运行截图



## 02255: 重建二叉树

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

思路：好久前的作业中类似题目的类似思路

代码

```

def front_mid_to_back(front, mid):
    if not mid:
        return ''

    idx = mid.index(front[0])
    left_front = front[1: idx + 1]
    right_front = front[idx + 1:]
    left_mid = mid[: idx]
    right_mid = mid[idx + 1:]

    return front_mid_to_back(left_front, left_mid) +
    front_mid_to_back(right_front, right_mid) + front[0]

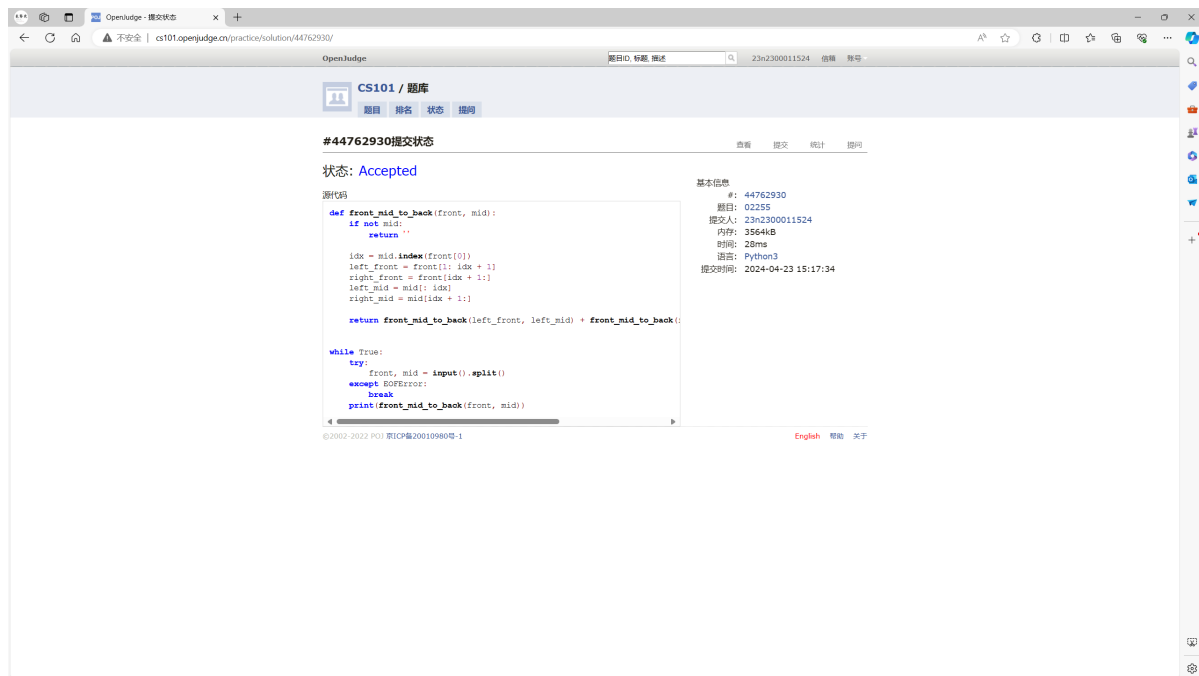
```

```

while True:
    try:
        front, mid = input().split()
    except EOFError:
        break
    print(front_mid_to_back(front, mid))

```

## 代码运行截图



## 01426: Find The Multiple

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

要求用bfs实现

思路：bfs打表

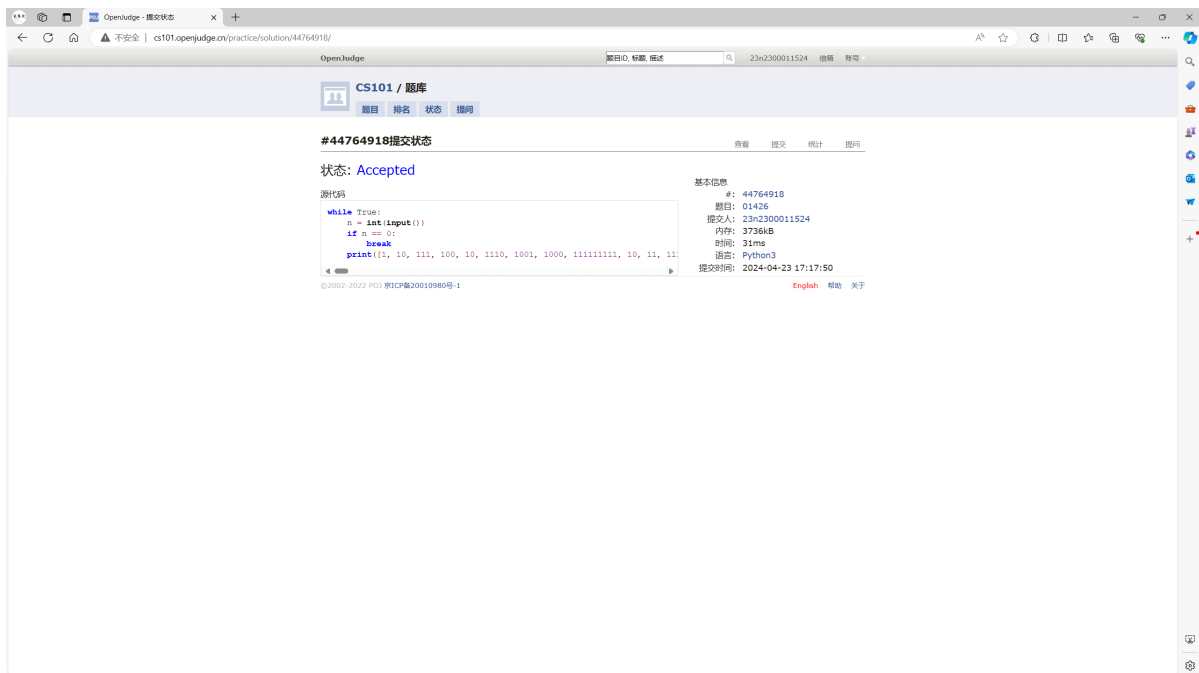
代码

```

while True:
    n = int(input())
    if n == 0:
        break
    print([1, 10, 111, 100, 10, 1110, 1001, 1000, 111111111, 10, 11, 11100, 1001,
10010, 1110, 10000, 11101, 1111111110, 11001, 100, 10101, 110, 110101, 111000,
100, 10010, 1101111111, 100100, 1101101, 1110, 111011, 100000, 111111, 111010,
10010, 11111111100, 111, 110010, 10101, 1000, 11111, 101010, 1101101, 1100,
11111111110, 1101010, 10011, 1110000, 1100001, 100, 100011, 100100, 100011,
11011111110, 110, 1001000, 11001, 11011010, 11011111, 11100, 100101, 1110110,
1111011111, 1000000, 10010, 1111110, 1101011, 1110100, 10000101, 10010, 10011,
111111111000, 10001, 1110, 11100, 1100100, 1001, 101010, 10010011, 10000,
1111111101, 111110, 101011, 1010100, 111010, 11011010, 11010111, 11000, 11010101,
1111111110, 1001, 11010100, 10000011, 100110, 110010, 11100000, 11100001,
11000010, 111111111111111111, 100, 101, 1000110, 11100001, 1001000, 101010,
1000110, 100010011, 110111111100, 1001010111, 110, 111, 10010000, 1011011,
110010, 1101010, 110110100, 10101111111, 110111110, 100111011, 111000, 11011,
1001010, 10001100111, 11101100, 1000, 11110111110, 11010011, 10000000, 100100001,
10010, 101001, 11111100, 11101111, 11010110, 11011111110, 11101000, 10001,
100001010, 110110101, 100100, 10011, 100110, 1001, 1111111110000, 11011010,
100010, 1100001, 11100, 110111, 11100, 1110001, 11001000, 10111110111, 10010,
1110110, 1010100, 10101101011, 100100110, 100011, 100000, 11101111, 11111111010,
1010111, 1111100, 1111110, 1010110, 11111011, 10101000, 10111101, 111010,
1111011111, 110110100, 1011001101, 110101110, 100100, 110000, 100101111,
110101010, 11010111, 11111111100, 1001111, 10010, 100101, 110101000, 1110,
100000110, 1001011, 1001100, 1010111010111, 110010, 11101111, 111000000, 11001,
111000010, 101010, 110000100, 1101000101, 1111111111111111110, 111000011, 1000][n
- 1])

```

代码运行截图



## 04115: 鸣人和佐助

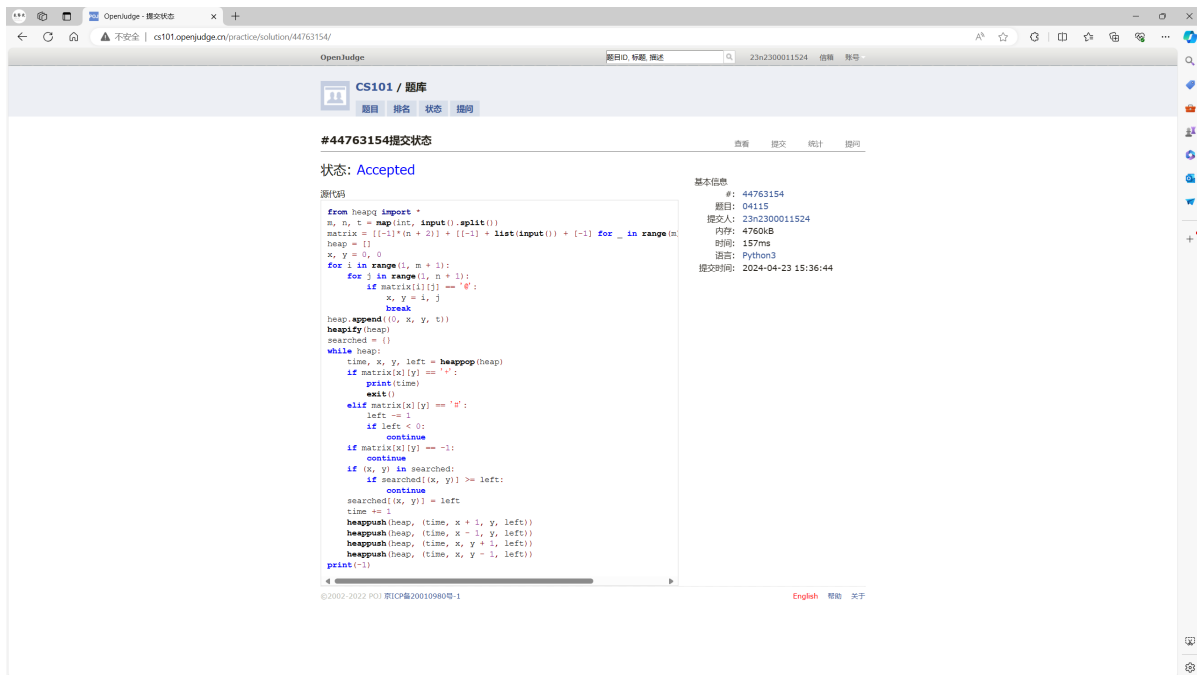
bfs, <http://cs101.openjudge.cn/practice/04115/>

思路: heap+bfs

代码

```
from heapq import *
m, n, t = map(int, input().split())
matrix = [[-1]*(n + 2)] + [[-1] + list(input()) + [-1] for _ in range(m)] +
[[-1]*(n + 2)]
heap = []
x, y = 0, 0
for i in range(1, m + 1):
    for j in range(1, n + 1):
        if matrix[i][j] == '@':
            x, y = i, j
            break
heap.append((0, x, y, t))
heapify(heap)
searched = {}
while heap:
    time, x, y, left = heappop(heap)
    if matrix[x][y] == '+':
        print(time)
        exit()
    elif matrix[x][y] == '#':
        left -= 1
        if left < 0:
            continue
    if matrix[x][y] == -1:
        continue
    if (x, y) in searched:
        if searched[(x, y)] >= left:
            continue
    searched[(x, y)] = left
    time += 1
    heappush(heap, (time, x + 1, y, left))
    heappush(heap, (time, x - 1, y, left))
    heappush(heap, (time, x, y + 1, left))
    heappush(heap, (time, x, y - 1, left))
print(-1)
```

代码运行截图



## 20106: 走山路

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

思路: bfs+heap

代码

```
import heapq

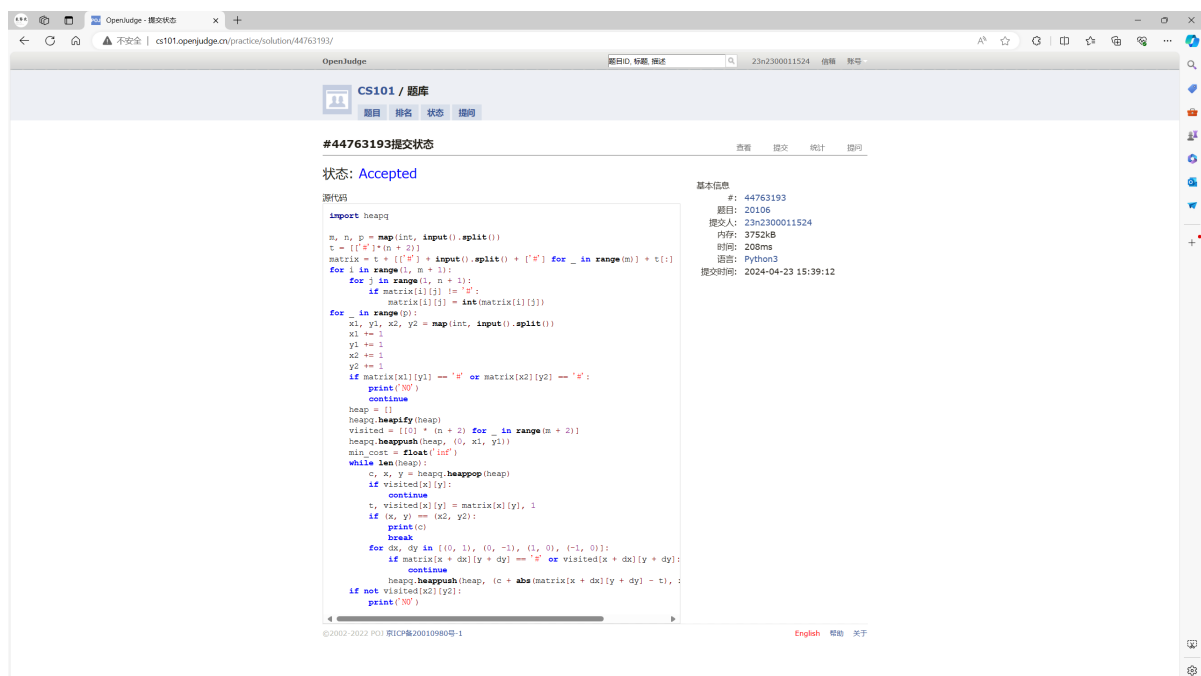
m, n, p = map(int, input().split())
t = [['#']*(n + 2)]
matrix = t + [['#'] + input().split() + ['#'] for _ in range(m)] + t[:]
for i in range(1, m + 1):
    for j in range(1, n + 1):
        if matrix[i][j] != '#':
            matrix[i][j] = int(matrix[i][j])
for _ in range(p):
    x1, y1, x2, y2 = map(int, input().split())
    x1 += 1
    y1 += 1
    x2 += 1
    y2 += 1
    if matrix[x1][y1] == '#' or matrix[x2][y2] == '#':
        print('NO')
        continue
    heap = []
    heapq.heapify(heap)
    visited = [[0] * (n + 2) for _ in range(m + 2)]
    heapq.heappush(heap, (0, x1, y1))
    min_cost = float('inf')
    while len(heap):
```

```

c, x, y = heapq.heappop(heap)
if visited[x][y]:
    continue
t, visited[x][y] = matrix[x][y], 1
if (x, y) == (x2, y2):
    print(c)
    break
for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
    if matrix[x + dx][y + dy] == '#' or visited[x + dx][y + dy]:
        continue
    heapq.heappush(heap, (c + abs(matrix[x + dx][y + dy] - t), x + dx, y
+ dy))
if not visited[x2][y2]:
    print('NO')

```

代码运行截图



## 05442: 兔子与星空

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

思路: kruskal 算法 (emm写完了才看见prim的提示)

代码

```

from heapq import *
from queue import Queue

n = int(input())
edges = {}
heap = []

```

```

for _ in range(n - 1):
    ls = input().split()
    idx = ls[0]
    edges[idx] = {}
    s, v = ls[2::2], ls[3::2]
    for i in range(int(ls[1])):
        heap.append((int(v[i]), idx, s[i]))
        if s[i] not in edges:
            edges[s[i]] = {}
heapify(heap)
ans = 0
while heap:
    value, a, b = heappop(heap)
    q = Queue()
    visited = {}
    q.put(a)
    circle = False
    while not q.empty():
        i = q.get()
        if i in visited:
            continue
        visited[i] = True
        if i == b:
            circle = True
        for j in edges[i]:
            q.put(j)
    if circle:
        continue
    edges[a][b] = edges[b][a] = value
    ans += value
print(ans)

```

## 代码运行截图

OpenJudge - 提交状态

CS101 / 题库

#44764633提交状态

状态: Accepted

源代码

```

from heapq import *
from queue import Queue

n = int(input())
edges = {}
heap = []
for _ in range(n - 1):
    ls = input().split()
    idx = ls[0]
    edges[idx] = {}
    s, v = ls[2::2], ls[3::2]
    for i in range(int(ls[1])):
        heap.append((int(v[i]), idx, s[i]))
        if s[i] not in edges:
            edges[s[i]] = {}
heapify(heap)
ans = 0
while heap:
    value, a, b = heappop(heap)
    q = Queue()
    visited = {}
    q.put(a)
    circle = False
    while not q.empty():
        i = q.get()
        if i in visited:
            continue
        visited[i] = True
        if i == b:
            circle = True
        for j in edges[i]:
            q.put(j)
    if circle:
        continue
    edges[a][b] = edges[b][a] = value
    ans += value
print(ans)

```

基本信息

#44764633  
题目: 05442  
提交人: 23n2300011524  
内存: 3772KB  
时间: 35ms  
语言: Python3  
提交时间: 2024-04-23 16:59:23

©2002-2022 POJ 赛ICP备20010980号-1

English 帮助 关于



## 2. 学习总结和收获

---

还可以。