Assignment #9: 图论：遍历，及 树算

Updated 1739 GMT+8 Apr 14, 2024

2024 spring, Complied 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

04081: 树的转换

http://cs101.openjudge.cn/dsapre/04081/

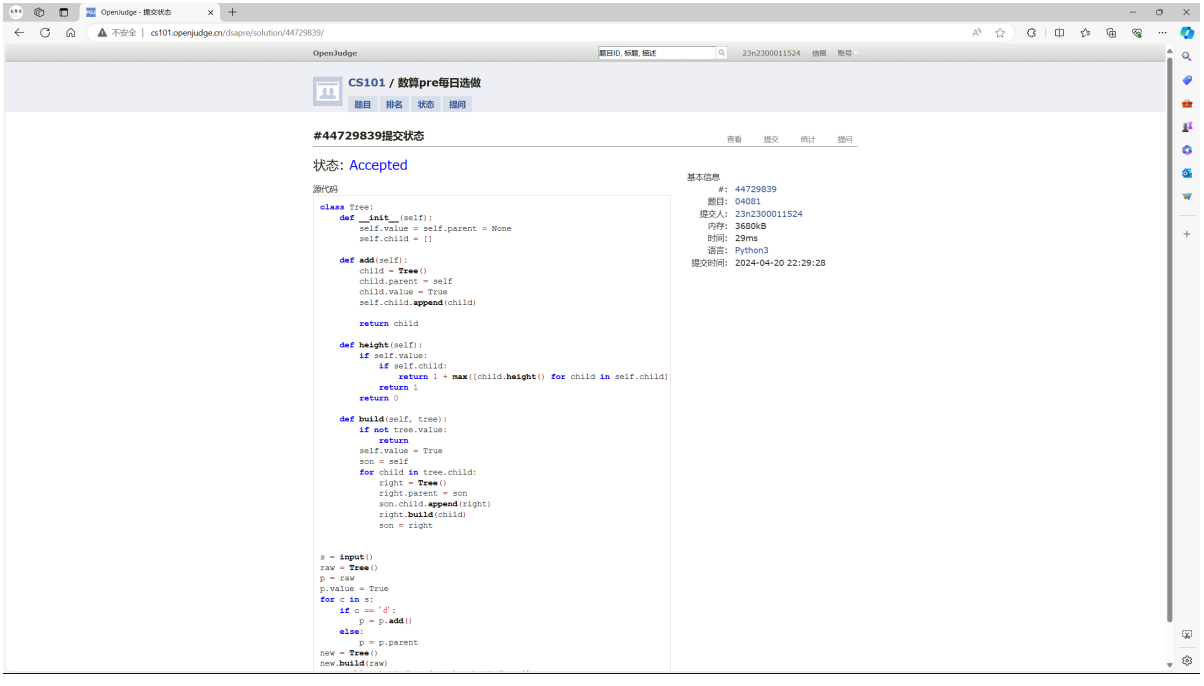
思路：正常做

代码

```
class Tree:
    def __init__(self):
        self.value = self.parent = None
        self.child = []

    def add(self):
        child = Tree()
        child.parent = self
        child.value = True
        self.child.append(child)

        return child

    def height(self):
```

```
        if self.value:
            if self.child:
                return 1 + max([child.height() for child in self.child])
            return 1
        return 0

    def build(self, tree):
        if not tree.value:
            return
        self.value = True
        son = self
        for child in tree.child:
            right = Tree()
            right.parent = son
            son.child.append(right)
            right.build(child)
            son = right


s = input()
raw = Tree()
p = raw
p.value = True
for c in s:
    if c == 'd':
        p = p.add()
    else:
        p = p.parent
new = Tree()
new.build(raw)
print(f'{raw.height() - 1} => {new.height() - 1}')
```

代码运行截图



08581: 扩展二叉树

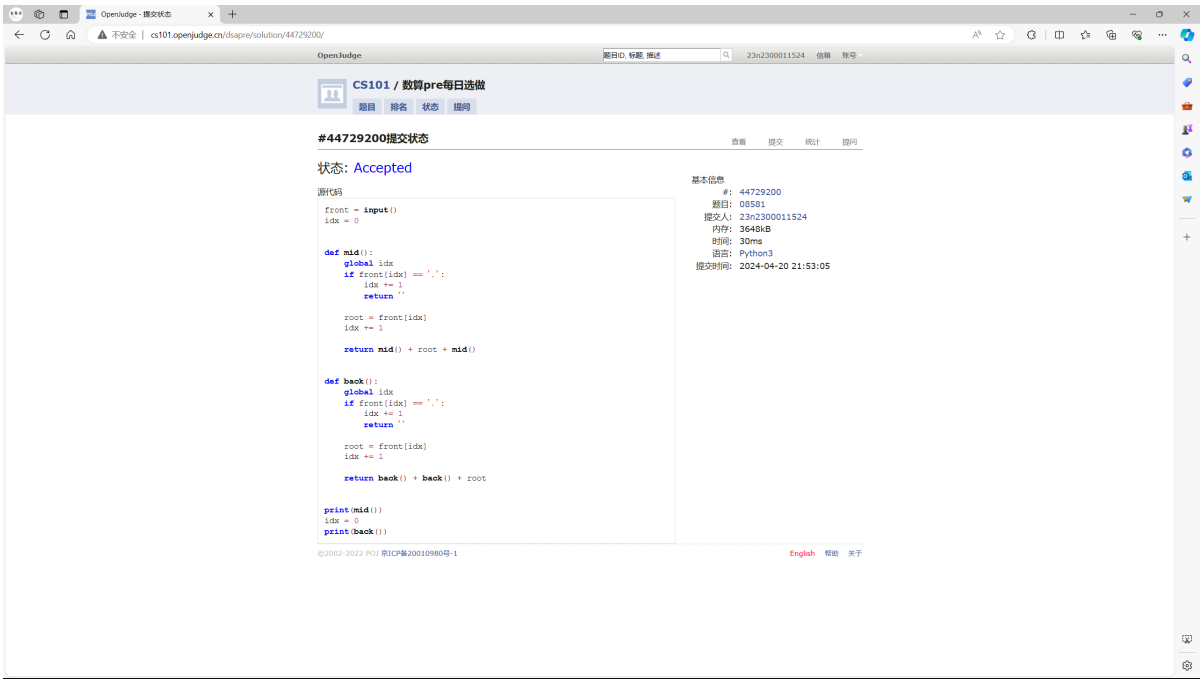
http://cs101.openjudge.cn/dsapre/08581/

思路：用好"."

代码

```
front = input()
idx = 0


def mid():
    global idx
    if front[idx] == '.':
        idx += 1
        return ''

    root = front[idx]
    idx += 1

    return mid() + root + mid()


def back():
    global idx
    if front[idx] == '.':
        idx += 1
        return ''

    root = front[idx]
    idx += 1

    return back() + back() + root


print(mid())
idx = 0
print(back())
```

代码运行截图



22067: 快速堆猪

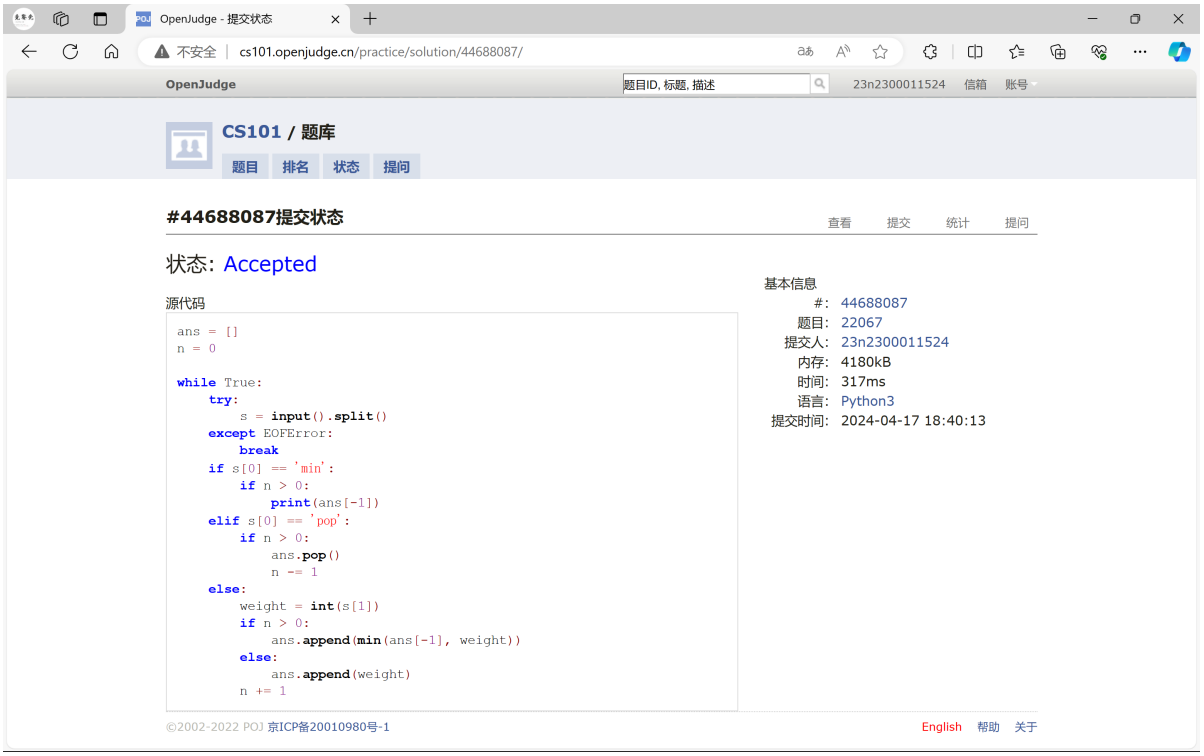
http://cs101.openjudge.cn/practice/22067/

思路：类dp

代码

```
ans = []
n = 0

while True:
    try:
        s = input().split()
    except EOFError:
        break
    if s[0] == 'min':
        if n > 0:
            print(ans[-1])
    elif s[0] == 'pop':
        if n > 0:
            ans.pop()
            n -= 1
    else:
        weight = int(s[1])
        if n > 0:
            ans.append(min(ans[-1], weight))
        else:
            ans.append(weight)
        n += 1
```

代码运行截图



04123: 马走日

dfs, http://cs101.openjudge.cn/practice/04123

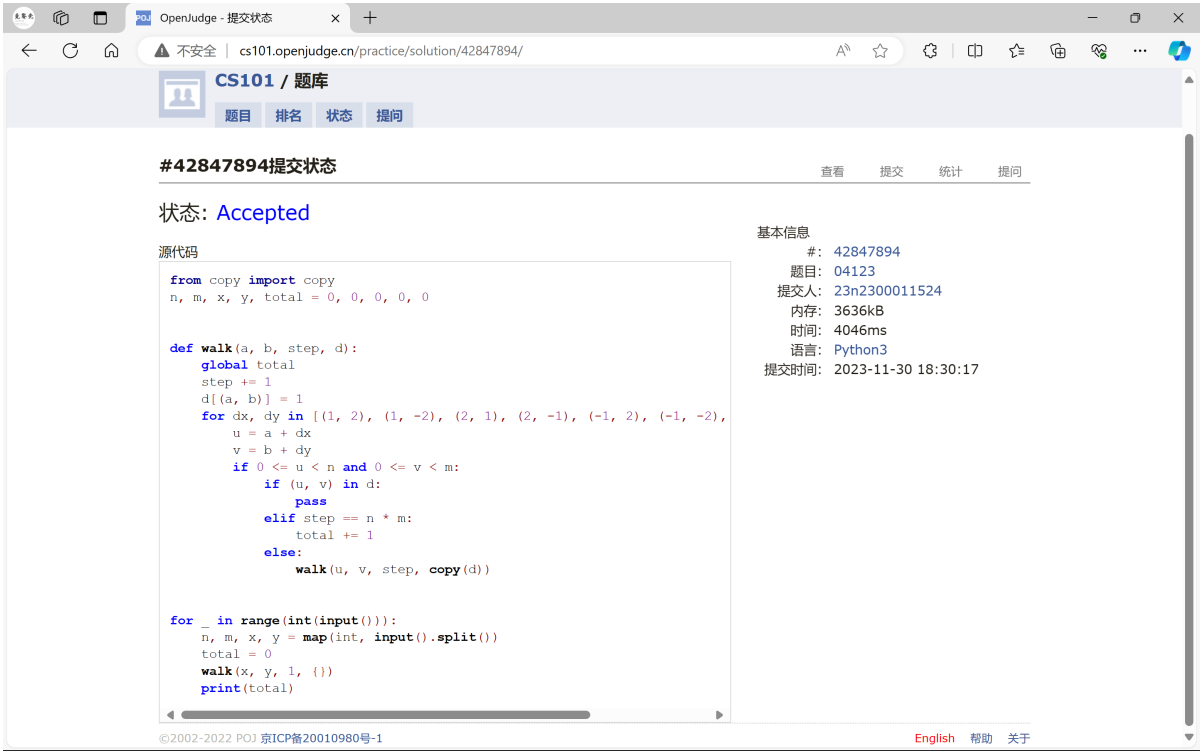
思路：dfs

代码

```
from copy import copy
n, m, x, y, total = 0, 0, 0, 0, 0


def walk(a, b, step, d):
    global total
    step += 1
    d[(a, b)] = 1
    for dx, dy in [(1, 2), (1, -2), (2, 1), (2, -1), (-1, 2), (-1, -2), (-2, 1),
(-2, -1)]:
        u = a + dx
        v = b + dy
        if 0 <= u < n and 0 <= v < m:
            if (u, v) in d:
                pass
            elif step == n * m:
                total += 1
            else:
                walk(u, v, step, copy(d))
```

```
for _ in range(int(input())):
    n, m, x, y = map(int, input().split())
    total = 0
    walk(x, y, 1, {})
    print(total)
```

代码运行截图



28046: 词梯

bfs, http://cs101.openjudge.cn/practice/28046/

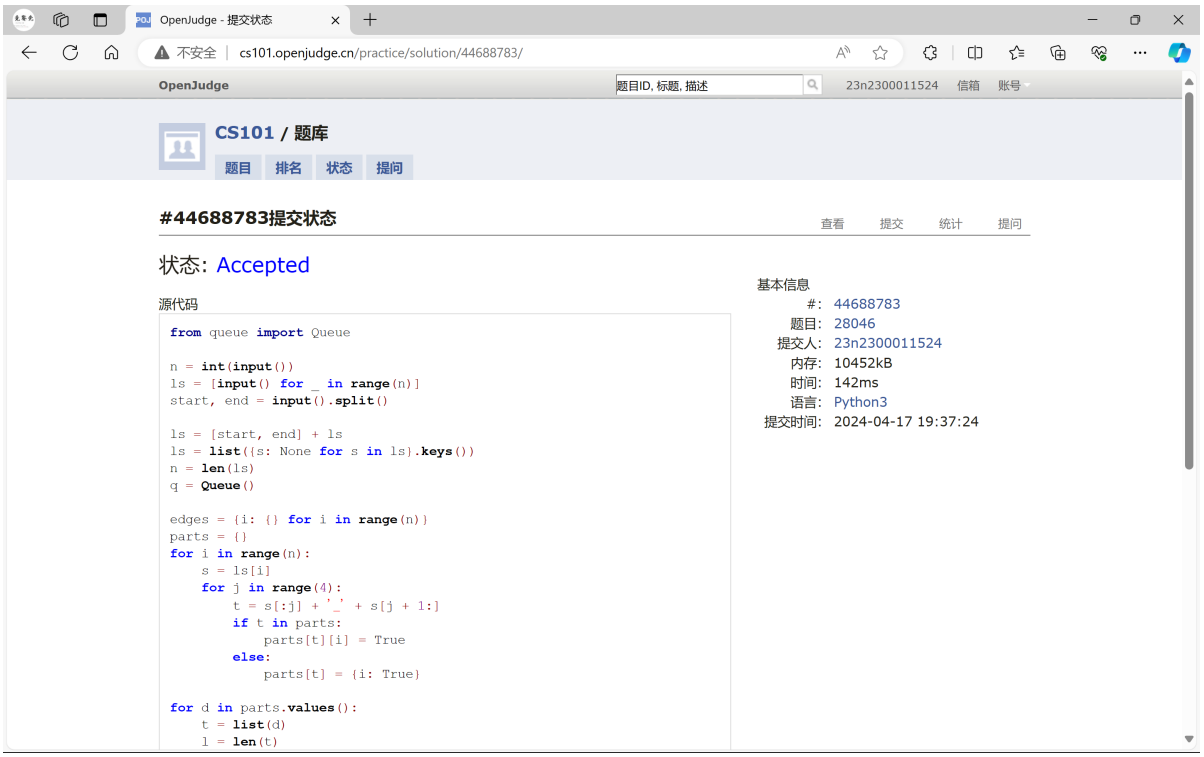
思路：标答思路

代码

```
from queue import Queue

n = int(input())
ls = [input() for _ in range(n)]
start, end = input().split()

ls = [start, end] + ls
ls = list({s: None for s in ls}.keys())
n = len(ls)
q = Queue()

edges = {i: {} for i in range(n)}
```

```
parts = {}
for i in range(n):
    s = ls[i]
    for j in range(4):
        t = s[:j] + '_' + s[j + 1:]
        if t in parts:
            parts[t][i] = True
        else:
            parts[t] = {i: True}

for d in parts.values():
    t = list(d)
    l = len(t)
    for i in range(l):
        for j in range(i + 1, l):
            edges[t[i]][t[j]] = True
            edges[t[j]][t[i]] = True

q.put([0])
visited = [False for i in range(n)]

while not q.empty():
    way = q.get()
    idx = way[-1]
    s = ls[idx]

    if idx == 1:
        print(' '.join([ls[i] for i in way]))
        exit()

    if visited[idx]:
        continue

    visited[idx] = True
    for j in edges[idx]:
        if not visited[j]:
            q.put(way + [j])
print('NO')
```

代码运行截图



28050: 骑士周游

dfs, http://cs101.openjudge.cn/practice/28050/

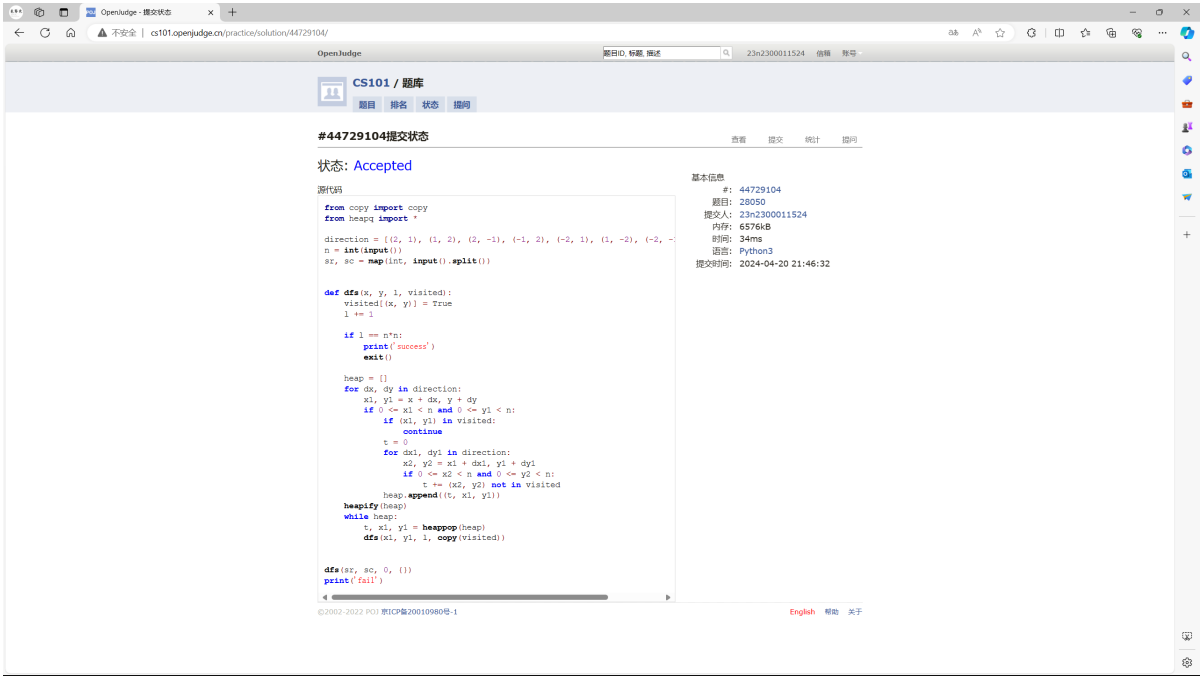
思路：Warnsdroff

代码

```
from copy import copy
from heapq import *

direction = [(2, 1), (1, 2), (2, -1), (-1, 2), (-2, 1), (1, -2), (-2, -1), (-1,
-2)]
n = int(input())
sr, sc = map(int, input().split())


def dfs(x, y, l, visited):
    visited[(x, y)] = True
    l += 1

    if l == n*n:
        print('success')
        exit()

    heap = []
    for dx, dy in direction:
        x1, y1 = x + dx, y + dy
        if 0 <= x1 < n and 0 <= y1 < n:
            if (x1, y1) in visited:
```

```
                continue
            t = 0
            for dx1, dy1 in direction:
                x2, y2 = x1 + dx1, y1 + dy1
                if 0 <= x2 < n and 0 <= y2 < n:
                    t += (x2, y2) not in visited
            heap.append((t, x1, y1))
    heapify(heap)
    while heap:
        t, x1, y1 = heappop(heap)
        dfs(x1, y1, l, copy(visited))


dfs(sr, sc, 0, {})
print('fail')
```

代码运行截图



2. 学习总结和收获

有点难度，最后一题一下子很难想到。期中结束，该补补进度了。