Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 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

19943: 图的拉普拉斯矩阵

matrices, http://cs101.openjudge.cn/practice/19943/

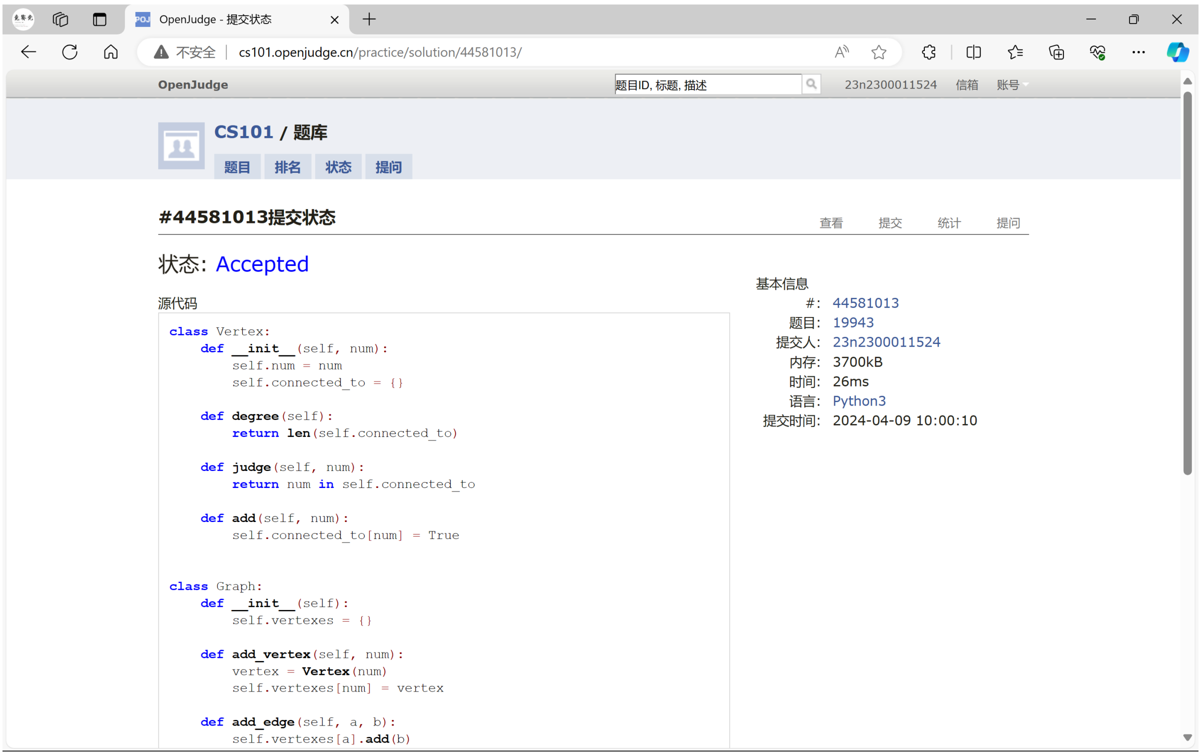
请定义Vertex类，Graph类，然后实现

思路：无

代码

```python
class Vertex:
    def __init__(self, num):
        self.num = num
        self.connected_to = {}

    def degree(self):
        return len(self.connected_to)

    def judge(self, num):
        return num in self.connected_to

    def add(self, num):
        self.connected_to[num] = True


class Graph:
    def __init__(self):
        self.vertexes = {}

    def add_vertex(self, num):
        vertex = Vertex(num)
        self.vertexes[num] = vertex

    def add_edge(self, a, b):
        self.vertexes[a].add(b)

    def judge(self, a, b):
        return self.vertexes[a].judge(b)


n, m = map(int, input().split())
graph = Graph()

for i in range(n):
    graph.add_vertex(i)

for _ in range(m):
    a, b = map(int, input().split())
    graph.add_edge(a, b)
    graph.add_edge(b, a)

l = [[0]*n for _ in range(n)]
for i in range(n):
    for j in range(n):
        l[i][j] = -graph.judge(i, j)
        if i == j:
            l[i][j] += graph.vertexes[i].degree()
for s in l:
    print(' '.join(map(str, s)))
```

代码运行截图



#44581013提交状态

状态: Accepted

源代码

```
class Vertex:
    def __init__(self, num):
        self.num = num
        self.connected_to = {}

    def degree(self):
        return len(self.connected_to)

    def judge(self, num):
        return num in self.connected_to

    def add(self, num):
        self.connected_to[num] = True


class Graph:
    def __init__(self):
        self.vertexes = {}

    def add_vertex(self, num):
        vertex = Vertex(num)
        self.vertexes[num] = vertex

    def add_edge(self, a, b):
        self.vertexes[a].add(b)
```

基本信息

#:	44581013
题目:	19943
提交人:	23n2300011524
内存:	3700kB
时间:	26ms
语言:	Python3
提交时间:	2024-04-09 10:00:10

18160: 最大连通域面积

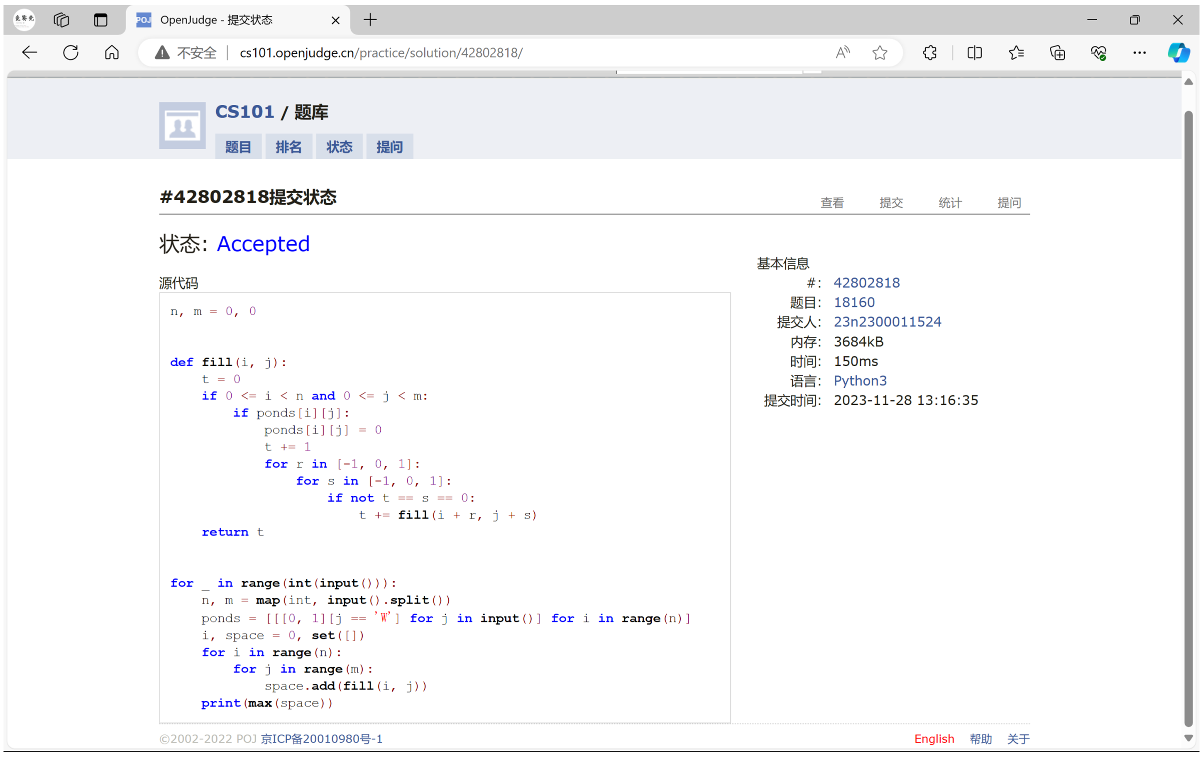
matrix/dfs similar, http://cs101.openjudge.cn/practice/18160

思路：dfs

代码

```
n, m = 0, 0


def fill(i, j):
    t = 0
    if 0 <= i < n and 0 <= j < m:
        if ponds[i][j]:
            ponds[i][j] = 0
            t += 1
            for r in [-1, 0, 1]:
                for s in [-1, 0, 1]:
                    if not t == s == 0:
                        t += fill(i + r, j + s)
    return t


for _ in range(int(input())):
    n, m = map(int, input().split())
    ponds = [[[0, 1][j == 'W'] for j in input()] for i in range(n)]
    i, space = 0, set([])
    for i in range(n):
        for j in range(m):
            space.add(fill(i, j))
    print(max(space))
```

代码运行截图


CS101 / 题库

#42802818提交状态

状态: Accepted

源代码

```
n, m = 0, 0


def fill(i, j):
    t = 0
    if 0 <= i < n and 0 <= j < m:
        if ponds[i][j]:
            ponds[i][j] = 0
            t += 1
            for r in [-1, 0, 1]:
                for s in [-1, 0, 1]:
                    if not t == s == 0:
                        t += fill(i + r, j + s)
    return t


for _ in range(int(input())):
    n, m = map(int, input().split())
    ponds = [[[0, 1][j == 'W'] for j in input()] for i in range(n)]
    i, space = 0, set([])
    for i in range(n):
        for j in range(m):
            space.add(fill(i, j))
    print(max(space))
```

基本信息

#:	42802818
题目:	18160
提交人:	23n2300011524
内存:	3684kB
时间:	150ms
语言:	Python3
提交时间:	2023-11-28 13:16:35

©2002-2022 POJ 京ICP备20010980号-1

sy383: 最大权值连通块

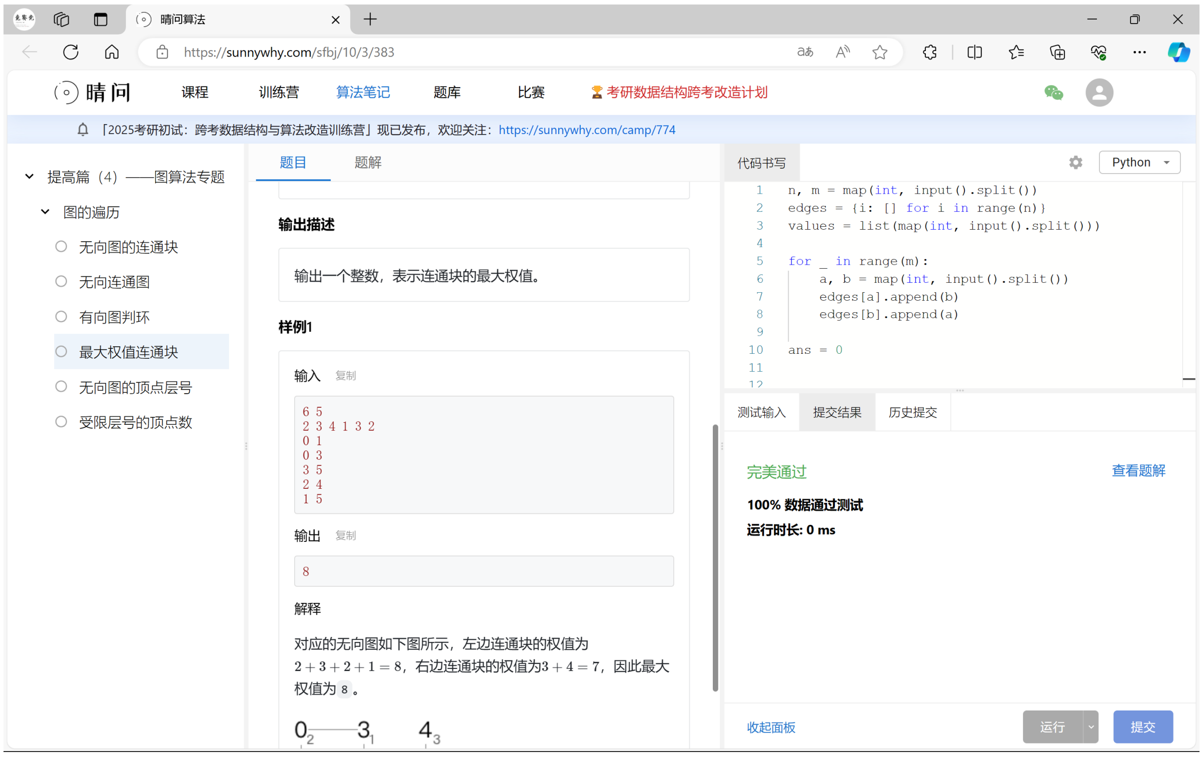
https://sunnywhy.com/sfbj/10/3/383

思路：dfs

代码

```
n, m = map(int, input().split())
edges = {i: [] for i in range(n)}
values = list(map(int, input().split()))

for _ in range(m):
    a, b = map(int, input().split())
    edges[a].append(b)
    edges[b].append(a)

ans = 0


def dfs(idx, visited):
    visited[idx] = True
    value = values[idx]
    for i in edges[idx]:
        if visited[i]:
            continue
        value += dfs(i, visited)
    return value


for i in range(n):
    ans = max(ans, dfs(i, [False] * n))
print(ans)
```

代码运行截图



晴问算法

https://sunnywhy.com/sfbj/10/3/383

晴问　课程　训练营　算法笔记　题库　比赛　考研数据结构跨考改造计划

「2025考研初试：跨考数据结构与算法改造训练营」现已发布，欢迎关注：https://sunnywhy.com/camp/774

提高篇（4）——图算法专题

图的遍历

- 无向图的连通块
- 无向连通图
- 有向图判环
- 最大权值连通块
- 无向图的顶点层号
- 受限层号的顶点数

题目　题解

输出描述

输出一个整数，表示连通块的最大权值。

样例1

输入　复制

```
6 5
2 3 4 1 3 2
0 1
0 3
3 5
2 4
1 5
```

输出　复制

```
8
```

解释

对应的无向图如下图所示，左边连通块的权值为 $2+3+2+1=8$，右边连通块的权值为$3+4=7$，因此最大权值为 8 。

代码书写　Python

```
n, m = map(int, input().split())
edges = {i: [] for i in range(n)}
values = list(map(int, input().split()))

for _ in range(m):
    a, b = map(int, input().split())
    edges[a].append(b)
    edges[b].append(a)

ans = 0
```

测试输入　提交结果　历史提交

完美通过　查看题解

100% 数据通过测试

运行时长: 0 ms

收起面板　运行　提交

03441: 4 Values whose Sum is 0

data structure/binary search, http://cs101.openjudge.cn/practice/03441

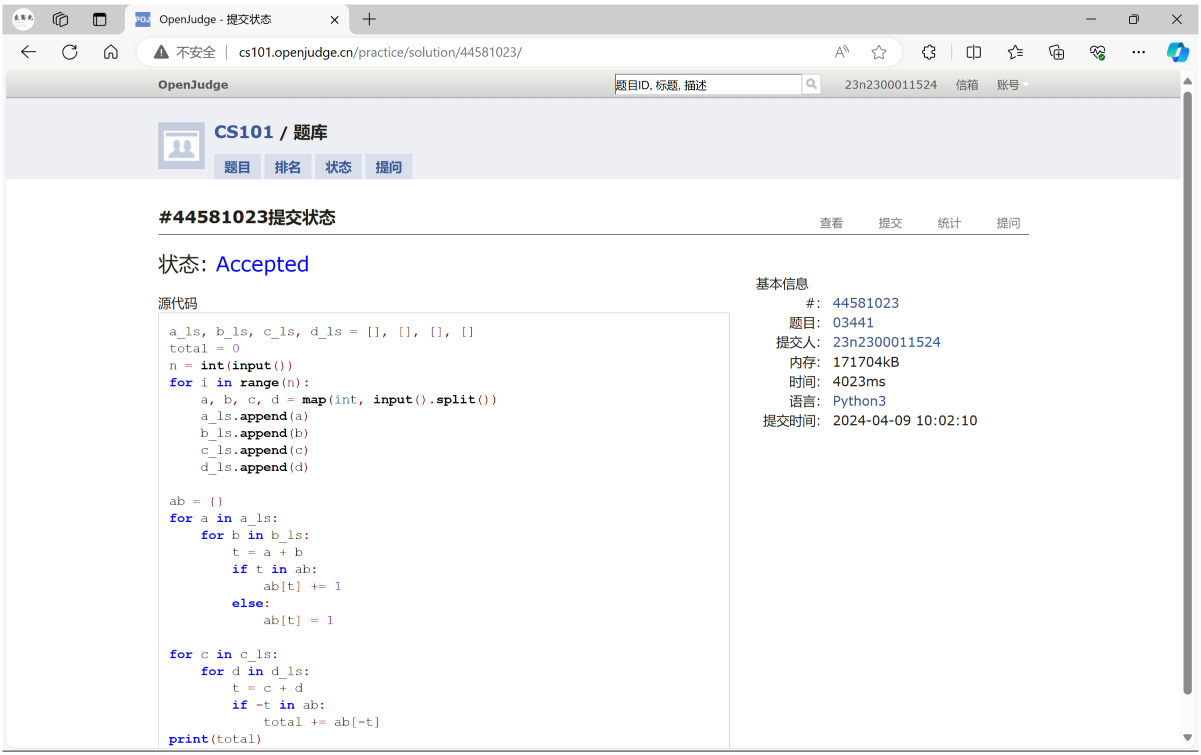
思路：桶，分别对a，b和c，d求解节约内存和时间

代码

```
a_ls, b_ls, c_ls, d_ls = [], [], [], []
total = 0
n = int(input())
for i in range(n):
    a, b, c, d = map(int, input().split())
    a_ls.append(a)
    b_ls.append(b)
    c_ls.append(c)
    d_ls.append(d)
```

```python
ab = {}
for a in a_ls:
    for b in b_ls:
        t = a + b
        if t in ab:
            ab[t] += 1
        else:
            ab[t] = 1

for c in c_ls:
    for d in d_ls:
        t = c + d
        if -t in ab:
            total += ab[-t]
print(total)
```

代码运行截图



#44581023提交状态

状态: Accepted

源代码

```python
a_ls, b_ls, c_ls, d_ls = [], [], [], []
total = 0
n = int(input())
for i in range(n):
    a, b, c, d = map(int, input().split())
    a_ls.append(a)
    b_ls.append(b)
    c_ls.append(c)
    d_ls.append(d)

ab = {}
for a in a_ls:
    for b in b_ls:
        t = a + b
        if t in ab:
            ab[t] += 1
        else:
            ab[t] = 1

for c in c_ls:
    for d in d_ls:
        t = c + d
        if -t in ab:
            total += ab[-t]
print(total)
```

基本信息

#:	44581023
题目:	03441
提交人:	23n2300011524
内存:	171704kB
时间:	4023ms
语言:	Python3
提交时间:	2024-04-09 10:02:10

04089: 电话号码

trie, http://cs101.openjudge.cn/practice/04089/

Trie 数据结构可能需要自学下。

思路：前缀字典/trie树

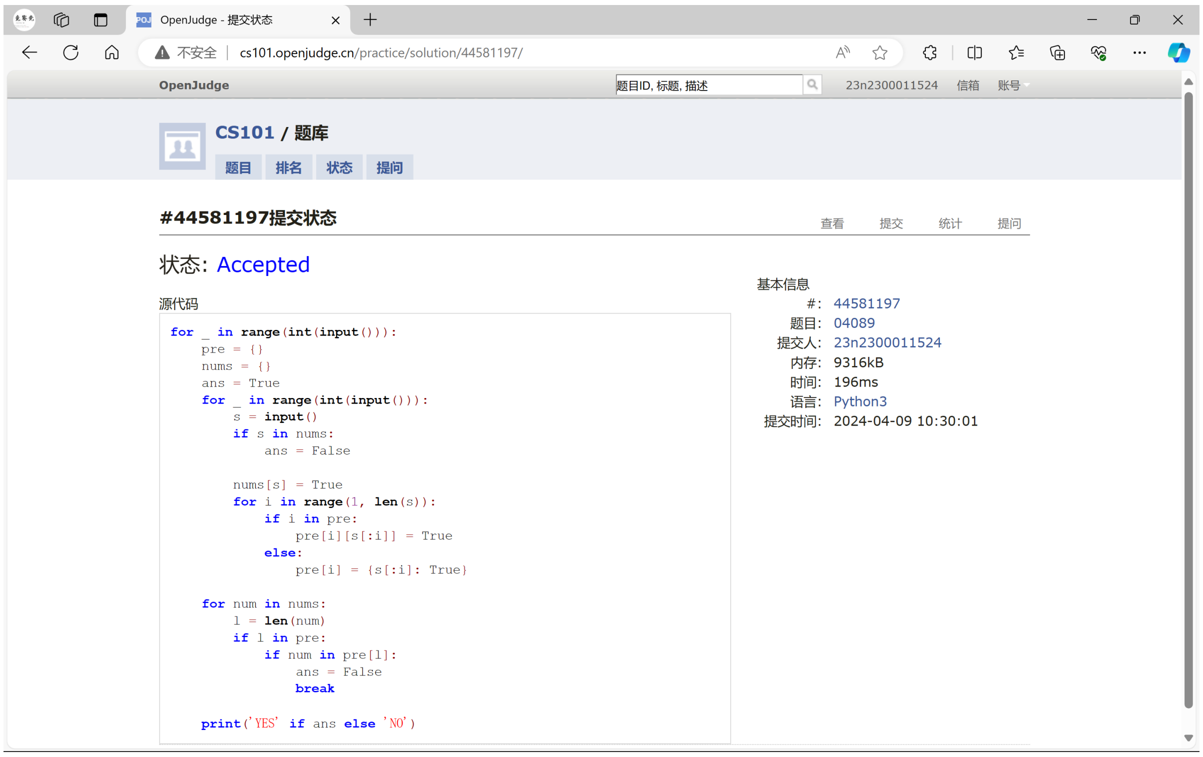
代码

```
#前缀字典
for _ in range(int(input())):
    pre = {}
    nums = {}
    ans = True
    for _ in range(int(input())):
        s = input()
        if s in nums:
            ans = False

        nums[s] = True
        for i in range(1, len(s)):
            if i in pre:
                pre[i][s[:i]] = True
            else:
                pre[i] = {s[:i]: True}

    for num in nums:
        l = len(num)
        if l in pre:
            if num in pre[l]:
                ans = False
                break

    print('YES' if ans else 'NO')
```

```
#Trie树
class Node:
    def __init__(self):
        self.child = {}


class TrieTree:
    def __init__(self):
        self.root = Node()

    def add(self, s):
        p = self.root
        for _ in range(len(s) - 1):
            t = s.pop()
            if t not in p.child:
                p.child[t] = Node()
            p = p.child[t]

    def judge(self, s):
        p = self.root
        ans = True
        for _ in range(len(s)):
            t = s.pop()
            if t in p.child:
                p = p.child[t]
            else:
                ans = False
                break
```

```python
        return ans


for _ in range(int(input())):
    trie = TrieTree()
    nums = {}
    ans = True
    for _ in range(int(input())):
        s = input()
        t = list(s)
        t.reverse()
        s = ''.join(t)
        if s in nums:
            ans = False

        nums[s] = True
        trie.add(t)

    for num in nums:
        if trie.judge(list(num)):
            ans = False
    print('YES' if ans else 'NO')
```

代码运行截图



OpenJudge - 提交状态

cs101.openjudge.cn/practice/solution/44581197/

CS101 / 题库

#44581197提交状态

状态: Accepted

源代码

```python
for _ in range(int(input())):
    pre = {}
    nums = {}
    ans = True
    for _ in range(int(input())):
        s = input()
        if s in nums:
            ans = False

        nums[s] = True
        for i in range(1, len(s)):
            if i in pre:
                pre[i][s[:i]] = True
            else:
                pre[i] = {s[:i]: True}

    for num in nums:
        l = len(num)
        if l in pre:
            if num in pre[l]:
                ans = False
                break

    print('YES' if ans else 'NO')
```

基本信息

#:	44581197
题目:	04089
提交人:	23n2300011524
内存:	9316kB
时间:	196ms
语言:	Python3
提交时间:	2024-04-09 10:30:01

04082: 树的镜面映射

http://cs101.openjudge.cn/practice/04082/

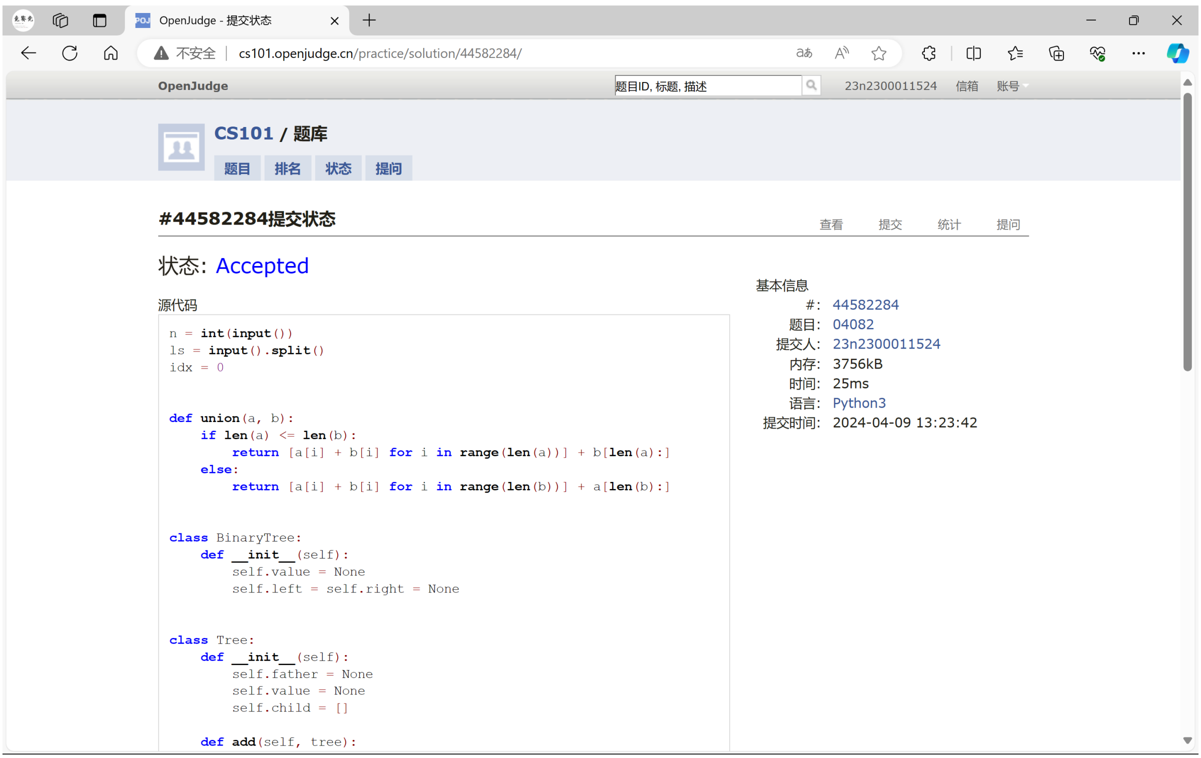
思路：建树

代码

```python
n = int(input())
ls = input().split()
idx = 0


def union(a, b):
    if len(a) <= len(b):
        return [a[i] + b[i] for i in range(len(a))] + b[len(a):]
    else:
        return [a[i] + b[i] for i in range(len(b))] + a[len(b):]


class BinaryTree:
    def __init__(self):
        self.value = None
        self.left = self.right = None


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

    def add(self, tree):
        self.child.append(tree)

    def level(self):
        ans = []
        for tree in self.child:
            ans = union(ans, tree.level())
        return [[self.value]] + ans


def build_binary_tree():
    global idx
    if idx >= len(ls):
        return None
    binary_tree = BinaryTree()
    if ls[idx][0] == '$':
        idx += 1
        return None

    binary_tree.value = ls[idx][0]
    if ls[idx][1] == '1':
        idx += 1
        return binary_tree

    idx += 1
    binary_tree.left = build_binary_tree()
    binary_tree.right = build_binary_tree()
    return binary_tree
```

```python
def build_tree(binary_tree, father):
    tree = Tree()
    tree.father = father
    if father:
        father.add(tree)
    tree.value = binary_tree.value
    if binary_tree.right:
        build_tree(binary_tree.right, tree.father)
    if binary_tree.left:
        build_tree(binary_tree.left, tree)
    return tree


binary_tree = build_binary_tree()
tree = build_tree(binary_tree, None)
ls = tree.level()
for i in range(len(ls)):
    ls[i].reverse()
    ls[i] = ' '.join(ls[i])
print(' '.join(ls))
```

代码运行截图



#44582284提交状态

状态: Accepted

源代码

```python
n = int(input())
ls = input().split()
idx = 0


def union(a, b):
    if len(a) <= len(b):
        return [a[i] + b[i] for i in range(len(a))] + b[len(a):]
    else:
        return [a[i] + b[i] for i in range(len(b))] + a[len(b):]


class BinaryTree:
    def __init__(self):
        self.value = None
        self.left = self.right = None


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

    def add(self, tree):
```

基本信息

#:	44582284
题目:	04082
提交人:	23n2300011524
内存:	3756kB
时间:	25ms
语言:	Python3
提交时间:	2024-04-09 13:23:42

2. 学习总结和收获

最后一题还是有点难度和复杂度的，期中季数算放一放

Trie树内存占用和时间明显比暴力散列表大?