Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Complied by 夏天明 元培学院

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：Windows 10 | 22H2

Python编程环境：Spyder IDE 5.4.3 | Python 3.11.4 64-bit

1. 题目

19943: 图的拉普拉斯矩阵

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

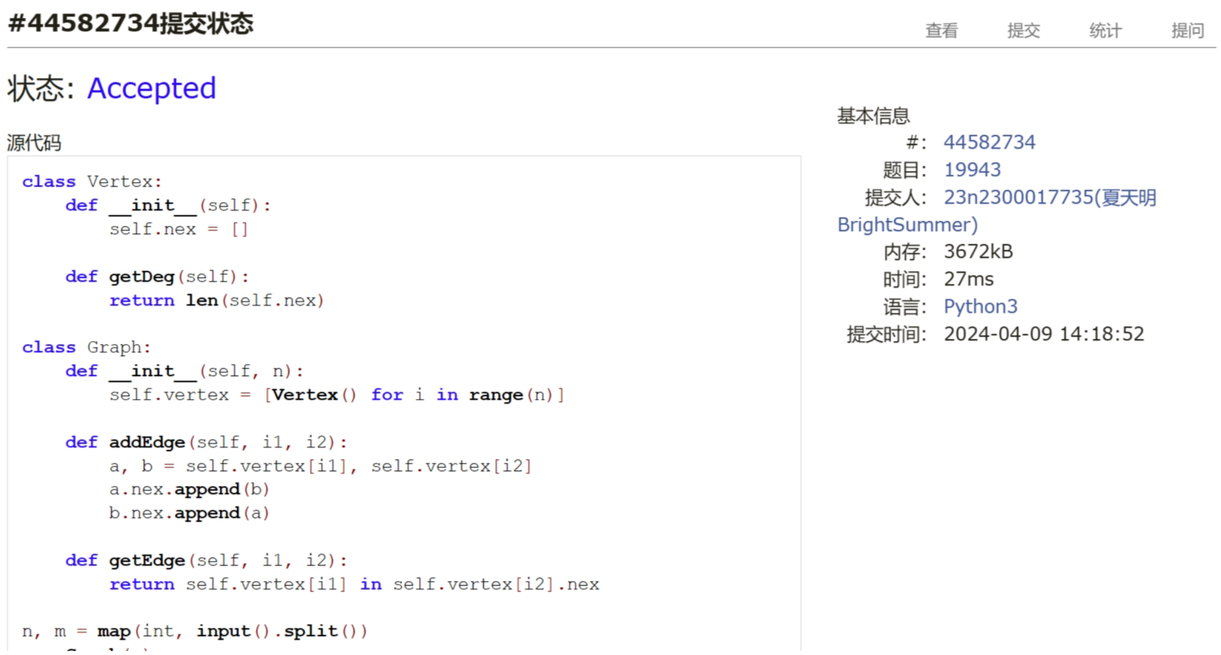
请定义Vertex类，Graph类，然后实现

思路：直接实现

代码

```python
class Vertex:
    def __init__(self):
        self.nex = []

    def getDeg(self):
        return len(self.nex)

class Graph:
    def __init__(self, n):
        self.vertex = [Vertex() for i in range(n)]
    def addEdge(self, i1, i2):
        a, b = self.vertex[i1], self.vertex[i2]
        a.nex.append(b)
        b.nex.append(a)

    def getEdge(self, i1, i2):
        return self.vertex[i1] in self.vertex[i2].nex

n, m = map(int, input().split())
g = Graph(n)
for o in range(m):
    g.addEdge(*map(int, input().split()))
for i in range(n):
    print(*[(0 if i != j else g.vertex[i].getDeg()) - g.getEdge(i, j) for j in range(n)])
```

代码运行截图



#44582734提交状态

状态: Accepted

源代码

```
class Vertex:
    def __init__(self):
        self.nex = []

    def getDeg(self):
        return len(self.nex)

class Graph:
    def __init__(self, n):
        self.vertex = [Vertex() for i in range(n)]

    def addEdge(self, i1, i2):
        a, b = self.vertex[i1], self.vertex[i2]
        a.nex.append(b)
        b.nex.append(a)

    def getEdge(self, i1, i2):
        return self.vertex[i1] in self.vertex[i2].nex

n, m = map(int, input().split())
```

基本信息

#: 44582734
题目: 19943
提交人: 23n2300017735(夏天明BrightSummer)
内存: 3672kB
时间: 27ms
语言: Python3
提交时间: 2024-04-09 14:18:52

18160: 最大连通域面积

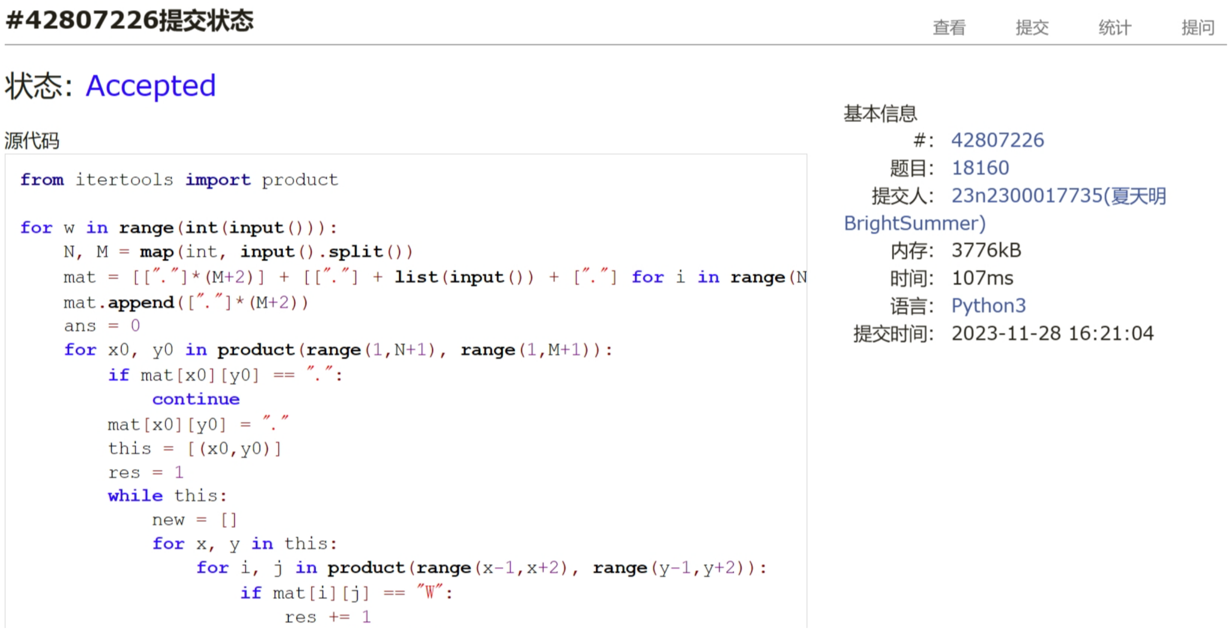
matrix/dfs similar, http://cs101.openjudge.cn/practice/18160

思路：bfs

代码

```python
from itertools import product

for w in range(int(input())):
    N, M = map(int, input().split())
    mat = [["."]*(M+2)] + [["."] + list(input()) + ["."] for i in range(N)]
    mat.append(["."]*(M+2))
    ans = 0
    for x0, y0 in product(range(1,N+1), range(1,M+1)):
        if mat[x0][y0] == ".":
            continue
        mat[x0][y0] = "."
        this = [(x0,y0)]
        res = 1
        while this:
            new = []
            for x, y in this:
                for i, j in product(range(x-1,x+2), range(y-1,y+2)):
                    if mat[i][j] == "W":
                        res += 1
                        mat[i][j] = "."
                        new.append((i,j))
            this = new
        ans = max(ans, res)
    print(ans)
```

代码运行截图



#42807226提交状态

状态: Accepted

源代码

```python
from itertools import product

for w in range(int(input())):
    N, M = map(int, input().split())
    mat = [["."]*(M+2)] + [["."] + list(input()) + ["."] for i in range(N
    mat.append(["."]*(M+2))
    ans = 0
    for x0, y0 in product(range(1,N+1), range(1,M+1)):
        if mat[x0][y0] == ".":
            continue
        mat[x0][y0] = "."
        this = [(x0,y0)]
        res = 1
        while this:
            new = []
            for x, y in this:
                for i, j in product(range(x-1,x+2), range(y-1,y+2)):
                    if mat[i][j] == "W":
                        res += 1
```

基本信息

#:	42807226
题目:	18160
提交人:	23n2300017735(夏天明 BrightSummer)
内存:	3776kB
时间:	107ms
语言:	Python3
提交时间:	2023-11-28 16:21:04

sy383: 最大权值连通块

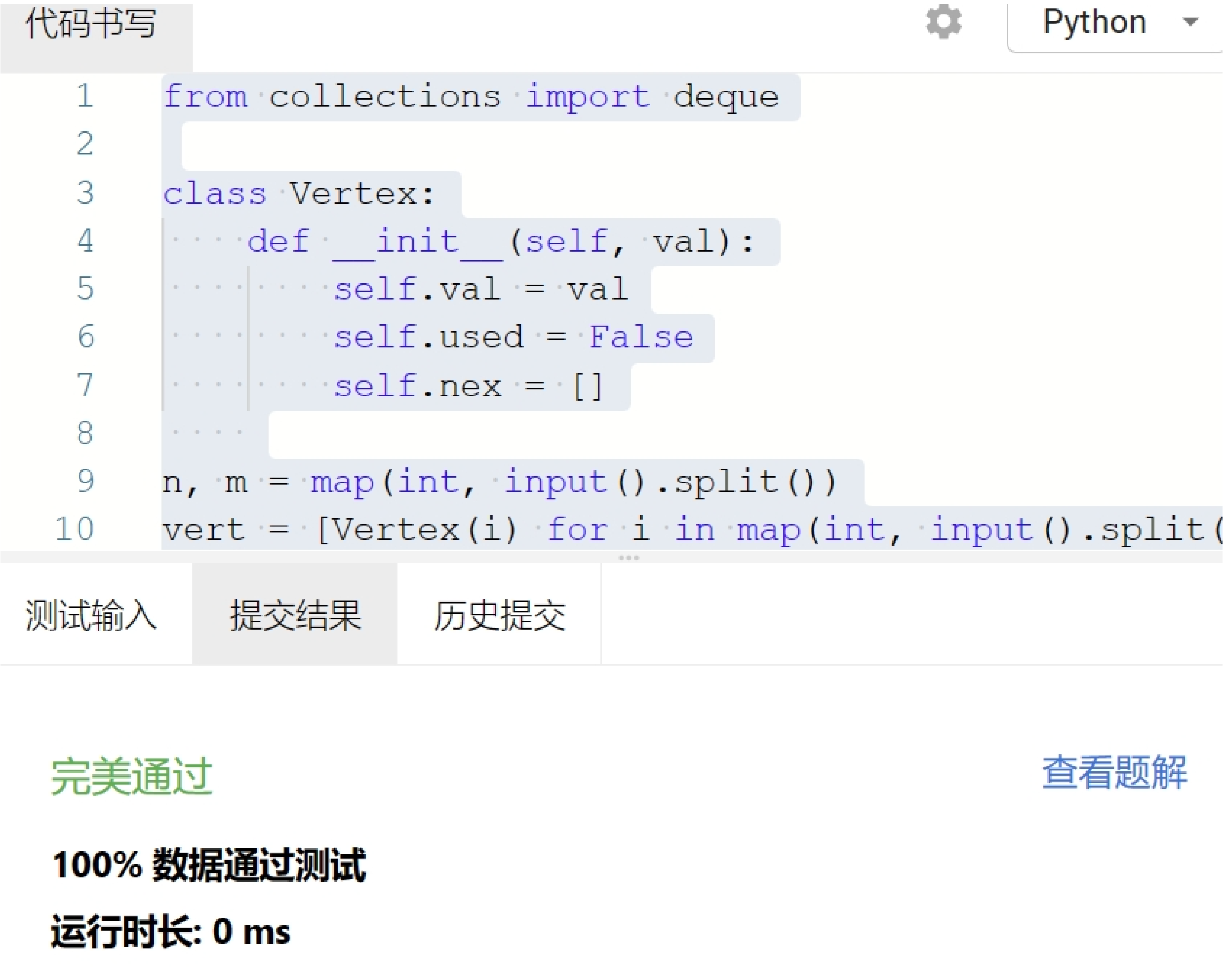
https://sunnywhy.com/sfbj/10/3/383

思路：bfs

代码

```
from collections import deque

class Vertex:
    def __init__(self, val):
        self.val = val
        self.used = False
        self.nex = []

n, m = map(int, input().split())
vert = [Vertex(i) for i in map(int, input().split())]
for o in range(m):
    a, b = map(int, input().split())
    vert[a].nex.append(vert[b])
    vert[b].nex.append(vert[a])
ans = 0
for v in vert:
    if not v.used:
        q = deque([v])
        res = 0
        while q:
            ve = q.popleft()
            if not ve.used:
                ve.used = True
                res += ve.val
                q.extend(ve.nex)
        ans = max(ans, res)
print(ans)
```

代码运行截图



```
from collections import deque

class Vertex:
    def __init__(self, val):
        self.val = val
        self.used = False
        self.nex = []
    
n, m = map(int, input().split())
vert = [Vertex(i) for i in map(int, input().split(
```

完美通过

100% 数据通过测试

运行时长: 0 ms

03441: 4 Values whose Sum is 0

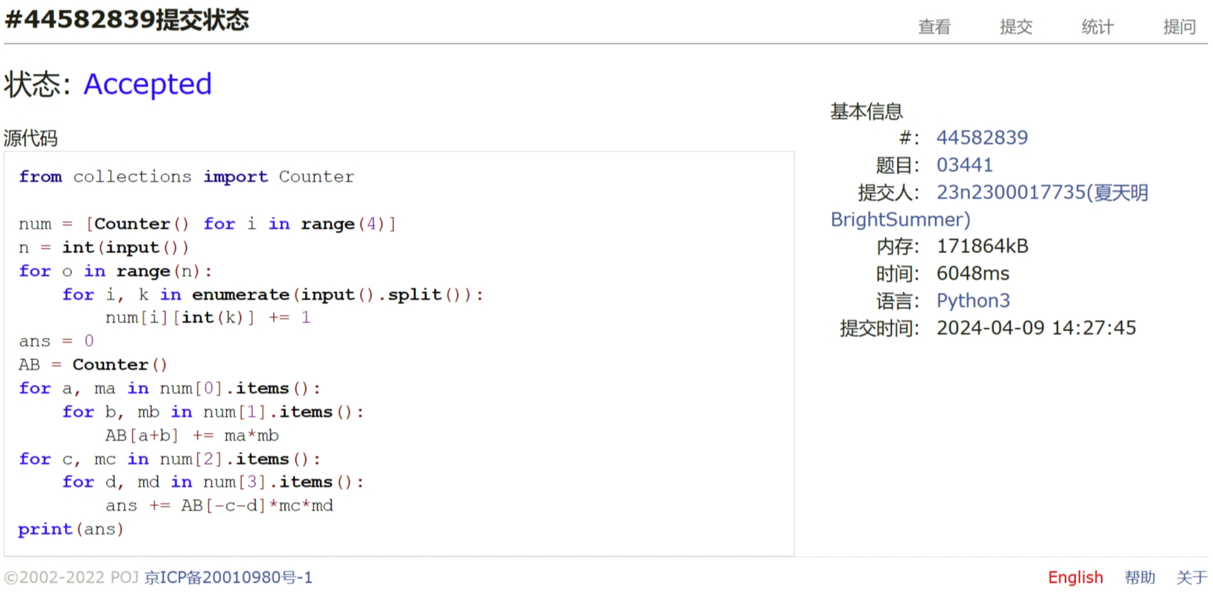
data structure/binary search, http://cs101.openjudge.cn/practice/03441

思路：先把AB加在一起，然后扫描cd

代码

```
from collections import Counter

num = [Counter() for i in range(4)]
n = int(input())
for o in range(n):
    for i, k in enumerate(input().split()):
        num[i][int(k)] += 1
ans = 0
AB = Counter()
for a, ma in num[0].items():
    for b, mb in num[1].items():
        AB[a+b] += ma*mb
for c, mc in num[2].items():
    for d, md in num[3].items():
        ans += AB[-c-d]*mc*md
print(ans)
```

代码运行截图



#44582839提交状态

状态: Accepted

源代码

```
from collections import Counter

num = [Counter() for i in range(4)]
n = int(input())
for o in range(n):
    for i, k in enumerate(input().split()):
        num[i][int(k)] += 1
ans = 0
AB = Counter()
for a, ma in num[0].items():
    for b, mb in num[1].items():
        AB[a+b] += ma*mb
for c, mc in num[2].items():
    for d, md in num[3].items():
        ans += AB[-c-d]*mc*md
print(ans)
```

基本信息

#:	44582839
题目:	03441
提交人:	23n2300017735(夏天明 BrightSummer)
内存:	171864kB
时间:	6048ms
语言:	Python3
提交时间:	2024-04-09 14:27:45

©2002-2022 POJ 京ICP备20010980号-1

04089: 电话号码

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

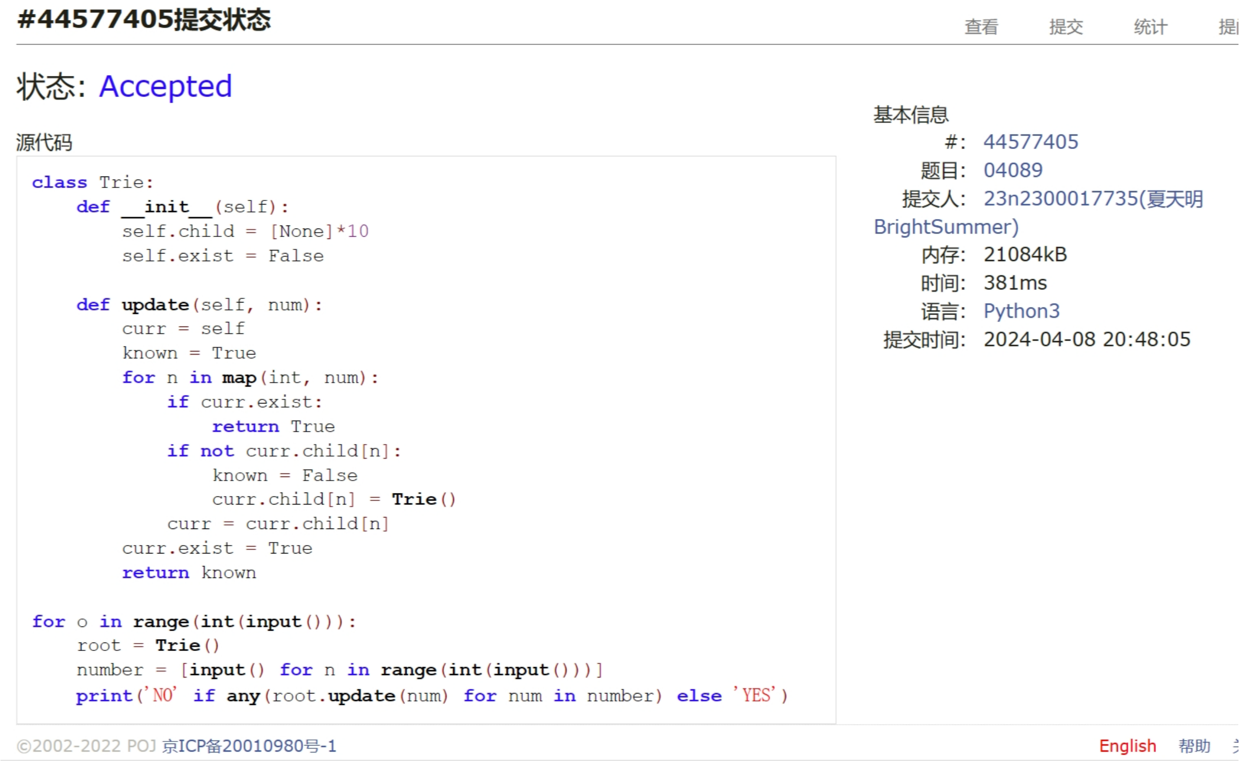
Trie 数据结构可能需要自学下。

思路：字典树

代码

```
class Trie:
    def __init__(self):
        self.child = [None]*10
        self.exist = False

    def update(self, num):
        curr = self
        known = True
        for n in map(int, num):
            if curr.exist:
                return True
            if not curr.child[n]:
                known = False
                curr.child[n] = Trie()
            curr = curr.child[n]
        curr.exist = True
        return known

for o in range(int(input())):
    root = Trie()
    number = [input() for n in range(int(input()))]
    print('NO' if any(root.update(num) for num in number) else 'YES')
```

代码运行截图



#44577405提交状态

状态: Accepted

源代码

```
class Trie:
    def __init__(self):
        self.child = [None]*10
        self.exist = False

    def update(self, num):
        curr = self
        known = True
        for n in map(int, num):
            if curr.exist:
                return True
            if not curr.child[n]:
                known = False
                curr.child[n] = Trie()
            curr = curr.child[n]
        curr.exist = True
        return known

for o in range(int(input())):
    root = Trie()
    number = [input() for n in range(int(input()))]
    print('NO' if any(root.update(num) for num in number) else 'YES')
```

基本信息
#: 44577405
题目: 04089
提交人: 23n2300017735(夏天明 BrightSummer)
内存: 21084kB
时间: 381ms
语言: Python3
提交时间: 2024-04-08 20:48:05

04082: 树的镜面映射

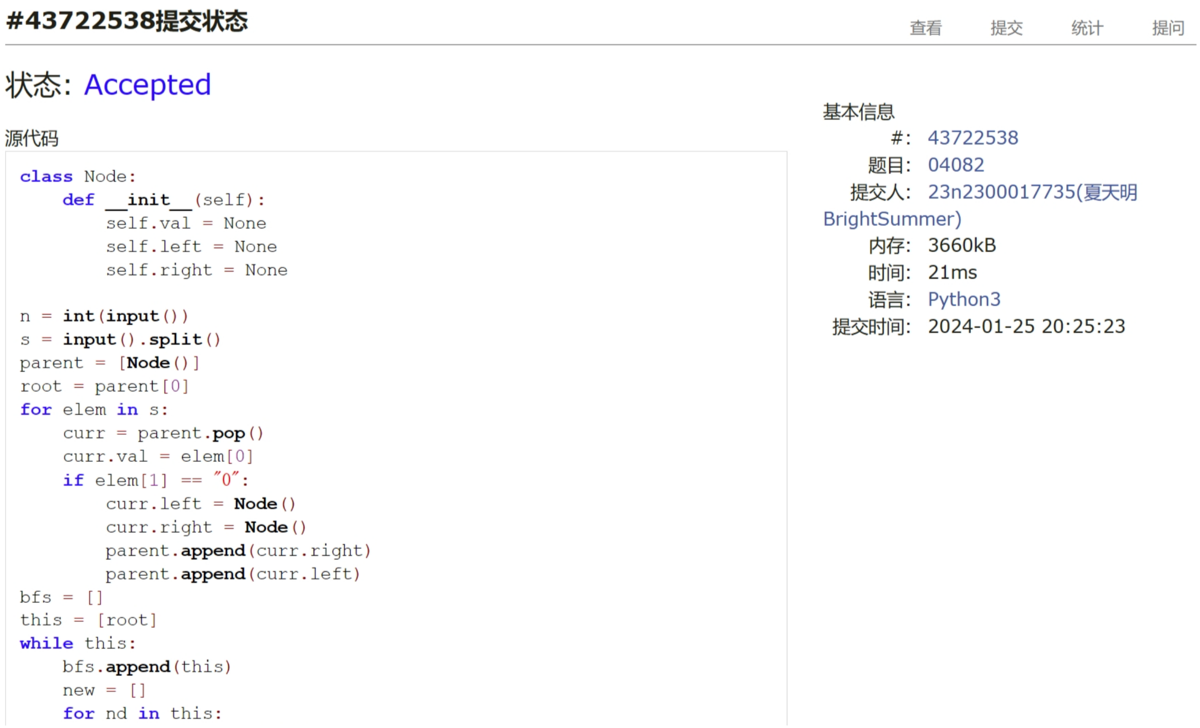
http://cs101.openjudge.cn/practice/04082/

思路：先用栈来处理输入的遍历序列，然后bfs层序遍历取反序就是所求

代码

```
class Node:
    def __init__(self):
        self.val = None
        self.left = None
        self.right = None

n = int(input())
s = input().split()
parent = [Node()]
root = parent[0]
for elem in s:
    curr = parent.pop()
    curr.val = elem[0]
    if elem[1] == "0":
        curr.left = Node()
        curr.right = Node()
        parent.append(curr.right)
        parent.append(curr.left)
bfs = []
this = [root]
while this:
    bfs.append(this)
    new = []
    for nd in this:
        if (child := nd.left):
            while child.val != "$":
                new.append(child)
                if not child.right:
                    break
                child = child.right
    this = new
ans = []
for row in bfs:
    ans.extend([nd.val for nd in row[::-1]])
print(*ans)
```

代码运行截图



#43722538提交状态

状态: Accepted

源代码

```
class Node:
    def __init__(self):
        self.val = None
        self.left = None
        self.right = None

n = int(input())
s = input().split()
parent = [Node()]
root = parent[0]
for elem in s:
    curr = parent.pop()
    curr.val = elem[0]
    if elem[1] == "0":
        curr.left = Node()
        curr.right = Node()
        parent.append(curr.right)
        parent.append(curr.left)
bfs = []
this = [root]
while this:
    bfs.append(this)
    new = []
    for nd in this:
```

基本信息

#: 43722538
题目: 04082
提交人: 23n2300017735(夏天明BrightSummer)
内存: 3660kB
时间: 21ms
语言: Python3
提交时间: 2024-01-25 20:25:23

2. 学习总结和收获

这次作业涉及到一些图的基础知识和树的拓展。有很多题在计算概论课程中已经学习过写法，但大都是特殊题目特殊处理，而在数算课程中，学习了树、图的知识，对这些题目就有了更深层的理解。例如电话号码题，之前可能只是利用python的sorted写一写，现在结合对Trie树的理解，可以思考采用何种排序方式最为高效等等。