Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Complied by ==同学的姓名、院系==

说明：

1）The complete process to learn DSA from scratch can be broken into 4 parts:

Learn about Time complexities, learn the basics of individual Data Structures, learn the basics of Algorithms, and practice Problems.

2）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。

3）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。

4）如果不能在截止前提交作业，请写明原因。

编程环境

==（请改为同学的操作系统、编程环境等）==

操作系统：macOS Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

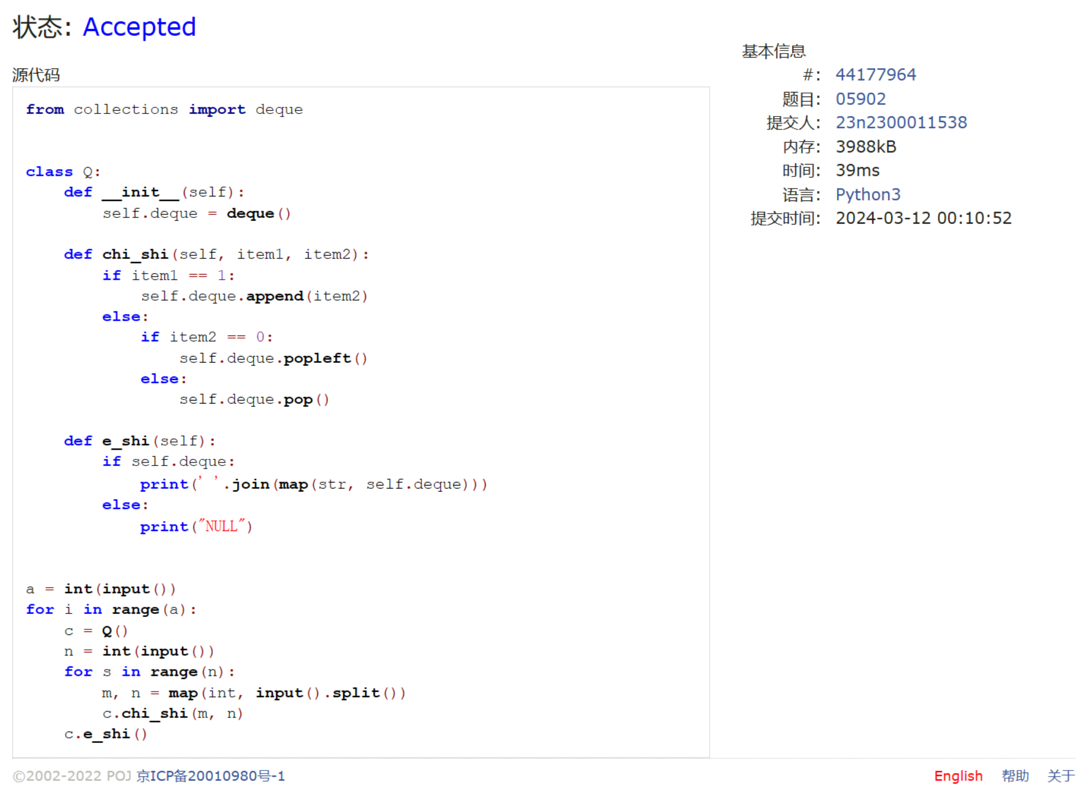
1. 题目

05902: 双端队列

http://cs101.openjudge.cn/practice/05902/

代码

```python
from collections import deque


class Q:
    def __init__(self):
        self.deque = deque()

    def chi_shi(self, item1, item2):
        if item1 == 1:
            self.deque.append(item2)
        else:
            if item2 == 0:
                self.deque.popleft()
            else:
                self.deque.pop()

    def e_shi(self):
        if self.deque:
            print(' '.join(map(str, self.deque)))
        else:
            print("NULL")


a = int(input())
for i in range(a):
    c = Q()
    n = int(input())
    for s in range(n):
        m, n = map(int, input().split())
        c.chi_shi(m, n)
    c.e_shi()
```



状态: Accepted

源代码

```python
from collections import deque


class Q:
    def __init__(self):
        self.deque = deque()

    def chi_shi(self, item1, item2):
        if item1 == 1:
            self.deque.append(item2)
        else:
            if item2 == 0:
                self.deque.popleft()
            else:
                self.deque.pop()

    def e_shi(self):
        if self.deque:
            print(' '.join(map(str, self.deque)))
        else:
            print("NULL")


a = int(input())
for i in range(a):
    c = Q()
    n = int(input())
    for s in range(n):
        m, n = map(int, input().split())
        c.chi_shi(m, n)
    c.e_shi()
```

基本信息

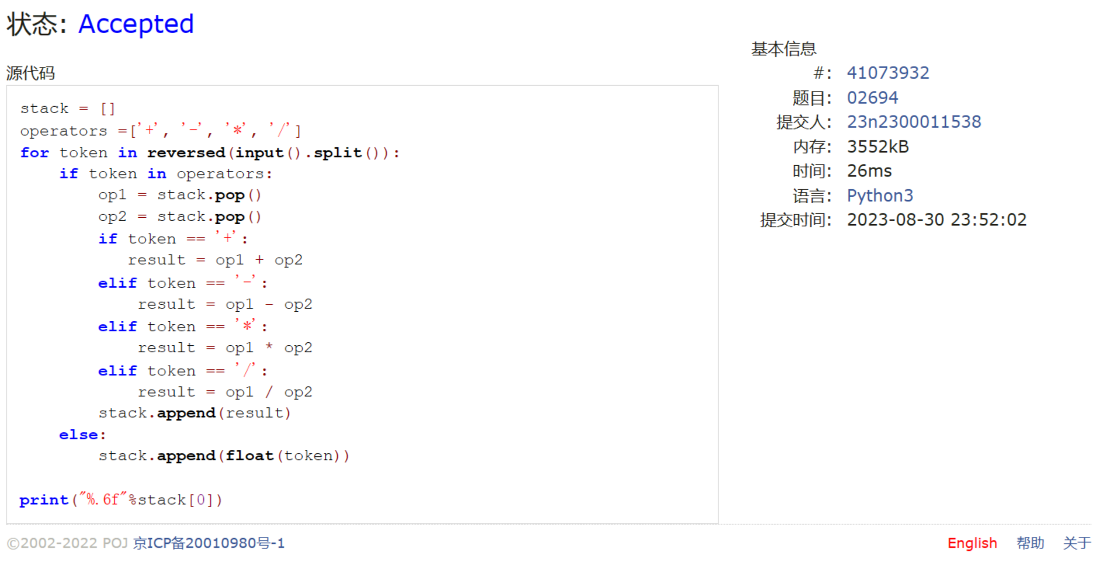
#:	44177964
题目:	05902
提交人:	23n2300011538
内存:	3988kB
时间:	39ms
语言:	Python3
提交时间:	2024-03-12 00:10:52

02694: 波兰表达式

http://cs101.openjudge.cn/practice/02694/

代码

```
stack = []
operators =['+', '-', '*', '/']
for token in reversed(input().split()):
    if token in operators:
        op1 = stack.pop()
        op2 = stack.pop()
        if token == '+':
           result = op1 + op2
        elif token == '-':
            result = op1 - op2
        elif token == '*':
            result = op1 * op2
        elif token == '/':
            result = op1 / op2
        stack.append(result)
    else:
        stack.append(float(token))

print("%.6f"%stack[0])
```



状态: Accepted

源代码

```
stack = []
operators =['+', '-', '*', '/']
for token in reversed(input().split()):
    if token in operators:
        op1 = stack.pop()
        op2 = stack.pop()
        if token == '+':
           result = op1 + op2
        elif token == '-':
            result = op1 - op2
        elif token == '*':
            result = op1 * op2
        elif token == '/':
            result = op1 / op2
        stack.append(result)
    else:
        stack.append(float(token))

print("%.6f"%stack[0])
```

基本信息

#:	41073932
题目:	02694
提交人:	23n2300011538
内存:	3552kB
时间:	26ms
语言:	Python3
提交时间:	2023-08-30 23:52:02

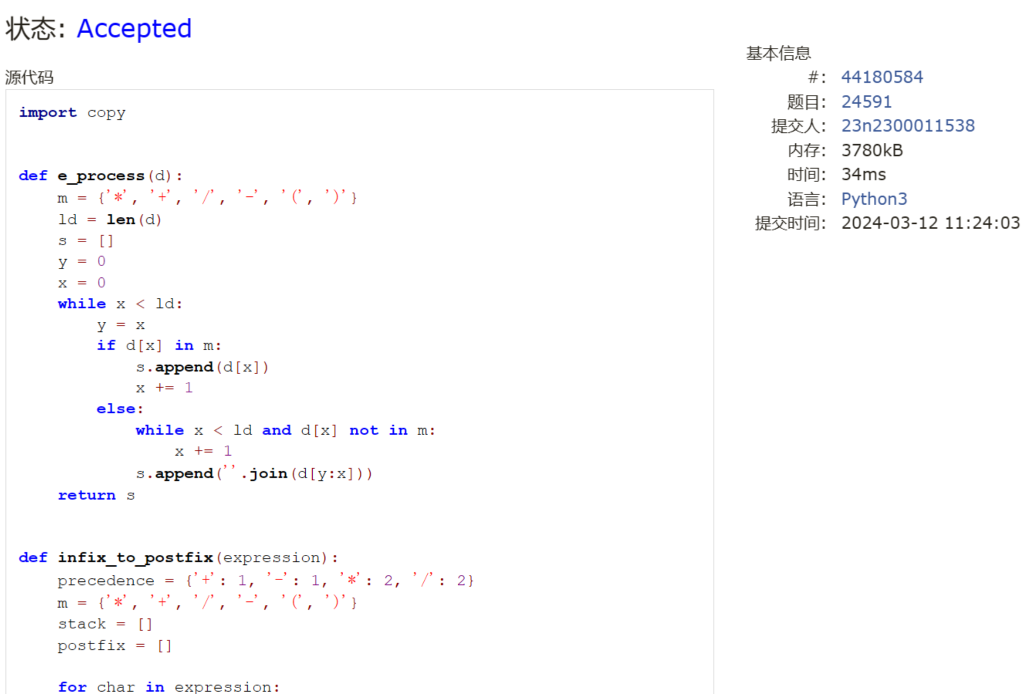
©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

24591: 中序表达式转后序表达式

http://cs101.openjudge.cn/practice/24591/

代码

```python
import copy


def e_process(d):
    m = {'*', '+', '/', '-', '(', ')'}
    ld = len(d)
    s = []
    y = 0
    x = 0
    while x < ld:
        y = x
        if d[x] in m:
            s.append(d[x])
            x += 1
        else:
            while x < ld and d[x] not in m:
                x += 1
            s.append(''.join(d[y:x]))
    return s


def infix_to_postfix(expression):
    precedence = {'+': 1, '-': 1, '*': 2, '/': 2}
    m = {'*', '+', '/', '-', '(', ')'}
    stack = []
    postfix = []

    for char in expression:
        if char not in m:
            postfix.append(char)
        elif char == '(':
            stack.append(char)
        elif char == ')':
            while stack and stack[-1] != '(':
                postfix.append(stack.pop())
            stack.pop()
        else:
            while stack and stack[-1] != '(' and precedence.get(stack[-1], 0) >=
precedence.get(char, 0):
                postfix.append(stack.pop())
            stack.append(char)

    while stack:
        postfix.append(stack.pop())

    for item in postfix:
        if item.strip():
            print(item, end=' ')
    print()


n = int(input())
for i in range(n):
    e = input()
    expression0 = [e[i] for i in range(len(e))]
    processed_expression = copy.deepcopy(e_process(expression0))
    infix_to_postfix(processed_expression)
```



状态: Accepted

源代码

```
import copy


def e_process(d):
    m = {'*', '+', '/', '-', '(', ')'}
    ld = len(d)
    s = []
    y = 0
    x = 0
    while x < ld:
        y = x
        if d[x] in m:
            s.append(d[x])
            x += 1
        else:
            while x < ld and d[x] not in m:
                x += 1
            s.append(''.join(d[y:x]))
    return s


def infix_to_postfix(expression):
    precedence = {'+': 1, '-': 1, '*': 2, '/': 2}
    m = {'*', '+', '/', '-', '(', ')'}
    stack = []
    postfix = []

    for char in expression:
```

基本信息

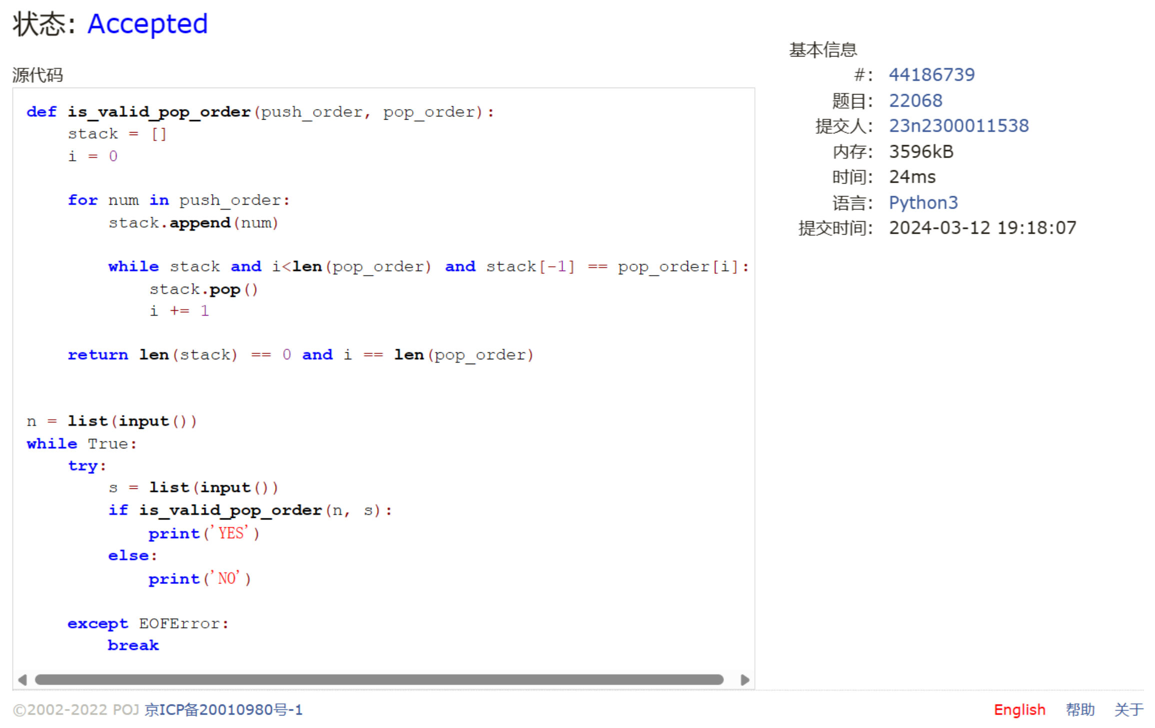
#:	44180584
题目:	24591
提交人:	23n2300011538
内存:	3780kB
时间:	34ms
语言:	Python3
提交时间:	2024-03-12 11:24:03

22068: 合法出栈序列

http://cs101.openjudge.cn/practice/22068/

代码

```python
def is_valid_pop_order(push_order, pop_order):
    stack = []
    i = 0

    for num in push_order:
        stack.append(num)

        while stack and i<len(pop_order) and stack[-1] == pop_order[i]:
            stack.pop()
            i += 1

    return len(stack) == 0 and i == len(pop_order)


n = list(input())
while True:
    try:
        s = list(input())
        if is_valid_pop_order(n, s):
            print('YES')
        else:
            print('NO')

    except EOFError:
        break
```


状态: Accepted

源代码

```
def is_valid_pop_order(push_order, pop_order):
    stack = []
    i = 0

    for num in push_order:
        stack.append(num)

        while stack and i<len(pop_order) and stack[-1] == pop_order[i]:
            stack.pop()
            i += 1

    return len(stack) == 0 and i == len(pop_order)


n = list(input())
while True:
    try:
        s = list(input())
        if is_valid_pop_order(n, s):
            print('YES')
        else:
            print('NO')

    except EOFError:
        break
```

基本信息

#:	44186739
题目:	22068
提交人:	23n2300011538
内存:	3596kB
时间:	24ms
语言:	Python3
提交时间:	2024-03-12 19:18:07

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

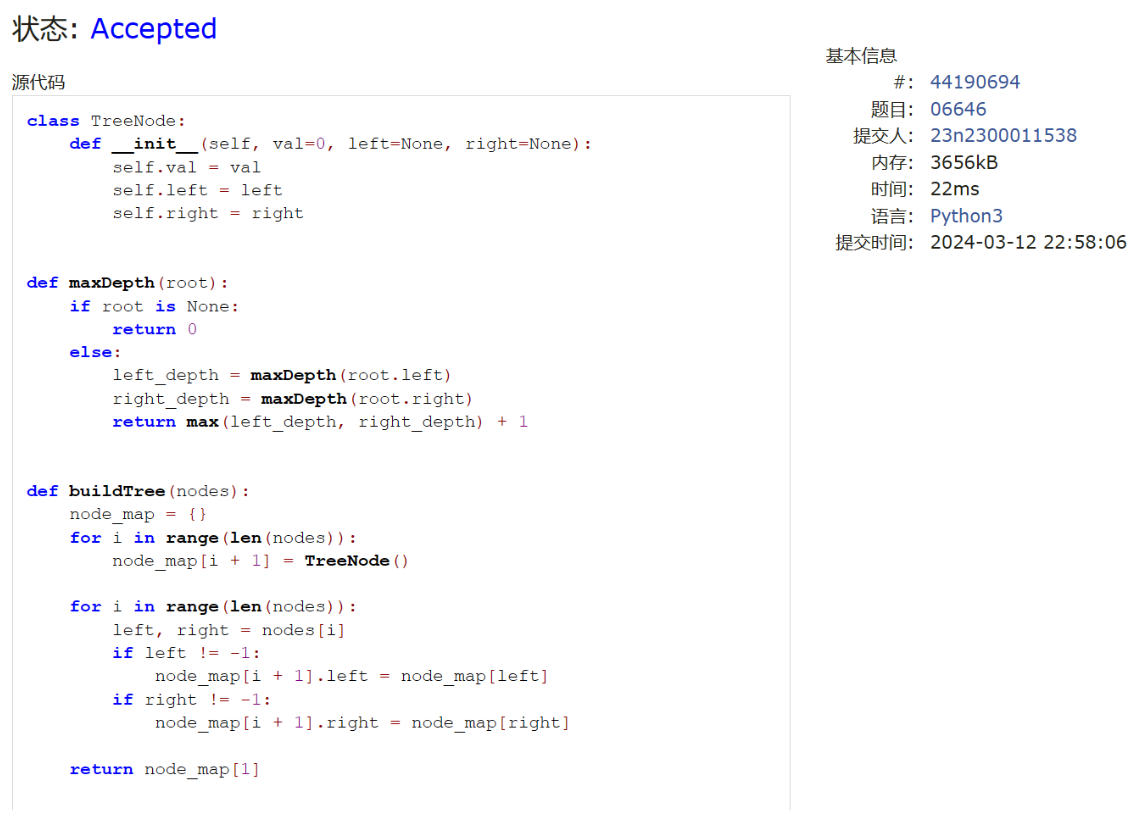
06646: 二叉树的深度

http://cs101.openjudge.cn/practice/06646/

代码

```
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right


def maxDepth(root):
    if root is None:
        return 0
    else:
        left_depth = maxDepth(root.left)
        right_depth = maxDepth(root.right)
        return max(left_depth, right_depth) + 1
```

```python
def buildTree(nodes):
    node_map = {}
    for i in range(len(nodes)):
        node_map[i + 1] = TreeNode()

    for i in range(len(nodes)):
        left, right = nodes[i]
        if left != -1:
            node_map[i + 1].left = node_map[left]
        if right != -1:
            node_map[i + 1].right = node_map[right]

    return node_map[1]


n = int(input())
nodes = []
for _ in range(n):
    left, right = map(int, input().split())
    nodes.append((left, right))

root = buildTree(nodes)
depth = maxDepth(root)
print(depth)
```



状态: Accepted

源代码

```python
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right


def maxDepth(root):
    if root is None:
        return 0
    else:
        left_depth = maxDepth(root.left)
        right_depth = maxDepth(root.right)
        return max(left_depth, right_depth) + 1


def buildTree(nodes):
    node_map = {}
    for i in range(len(nodes)):
        node_map[i + 1] = TreeNode()

    for i in range(len(nodes)):
        left, right = nodes[i]
        if left != -1:
            node_map[i + 1].left = node_map[left]
        if right != -1:
            node_map[i + 1].right = node_map[right]

    return node_map[1]
```

基本信息

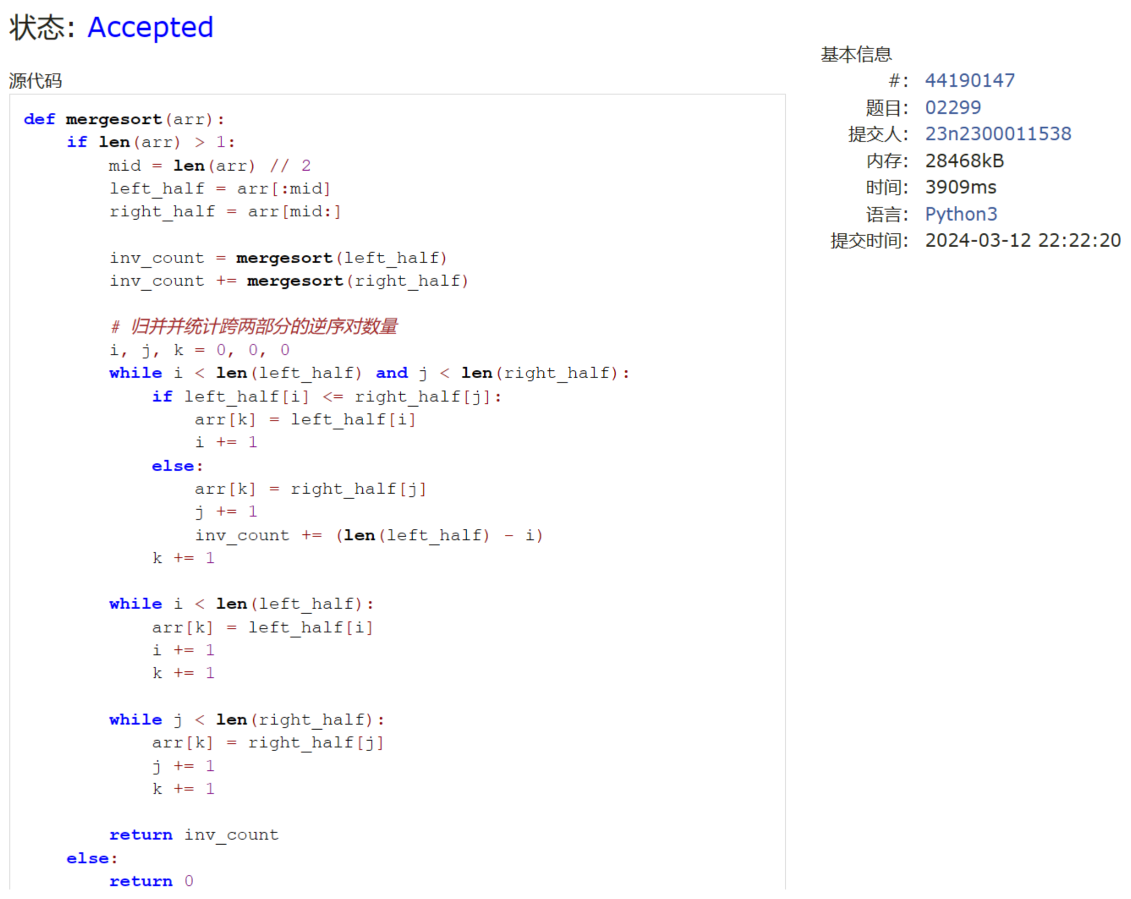
#:	44190694
题目:	06646
提交人:	23n2300011538
内存:	3656kB
时间:	22ms
语言:	Python3
提交时间:	2024-03-12 22:58:06

02299: Ultra-QuickSort

http://cs101.openjudge.cn/practice/02299/

代码

```python
def mergesort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        left_half = arr[:mid]
        right_half = arr[mid:]

        inv_count = mergesort(left_half)
        inv_count += mergesort(right_half)

        # 归并并统计跨两部分的逆序对数量
        i, j, k = 0, 0, 0
        while i < len(left_half) and j < len(right_half):
            if left_half[i] <= right_half[j]:
                arr[k] = left_half[i]
                i += 1
            else:
                arr[k] = right_half[j]
                j += 1
                inv_count += (len(left_half) - i)
            k += 1

        while i < len(left_half):
            arr[k] = left_half[i]
            i += 1
            k += 1

        while j < len(right_half):
            arr[k] = right_half[j]
            j += 1
            k += 1

        return inv_count
    else:
        return 0


while True:
    n = int(input())
    if n == 0:
        break
    original_list = []
    for i in range(n):
        original_list.append(int(input()))
    print(mergesort(original_list))
```



状态: Accepted

源代码

```python
def mergesort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        left_half = arr[:mid]
        right_half = arr[mid:]

        inv_count = mergesort(left_half)
        inv_count += mergesort(right_half)

        # 归并并统计跨两部分的逆序对数量
        i, j, k = 0, 0, 0
        while i < len(left_half) and j < len(right_half):
            if left_half[i] <= right_half[j]:
                arr[k] = left_half[i]
                i += 1
            else:
                arr[k] = right_half[j]
                j += 1
                inv_count += (len(left_half) - i)
            k += 1

        while i < len(left_half):
            arr[k] = left_half[i]
            i += 1
            k += 1

        while j < len(right_half):
            arr[k] = right_half[j]
            j += 1
            k += 1

        return inv_count
    else:
        return 0
```

基本信息

#:	44190147
题目:	02299
提交人:	23n2300011538
内存:	28468kB
时间:	3909ms
语言:	Python3
提交时间:	2024-03-12 22:22:20

2. 学习总结和收获

中序转后序debug了好久

上周任务没有完成呜呜呜

这周继续努力

遭遇了一些情感问题需要好好调整一下（）