Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Complied by ==陈奕妤 工学院==

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 Sonoma 14.4 (23E214)

Python编程环境：PyCharm 2023.3.1 (Professional Edition)

1. 题目

05902: 双端队列

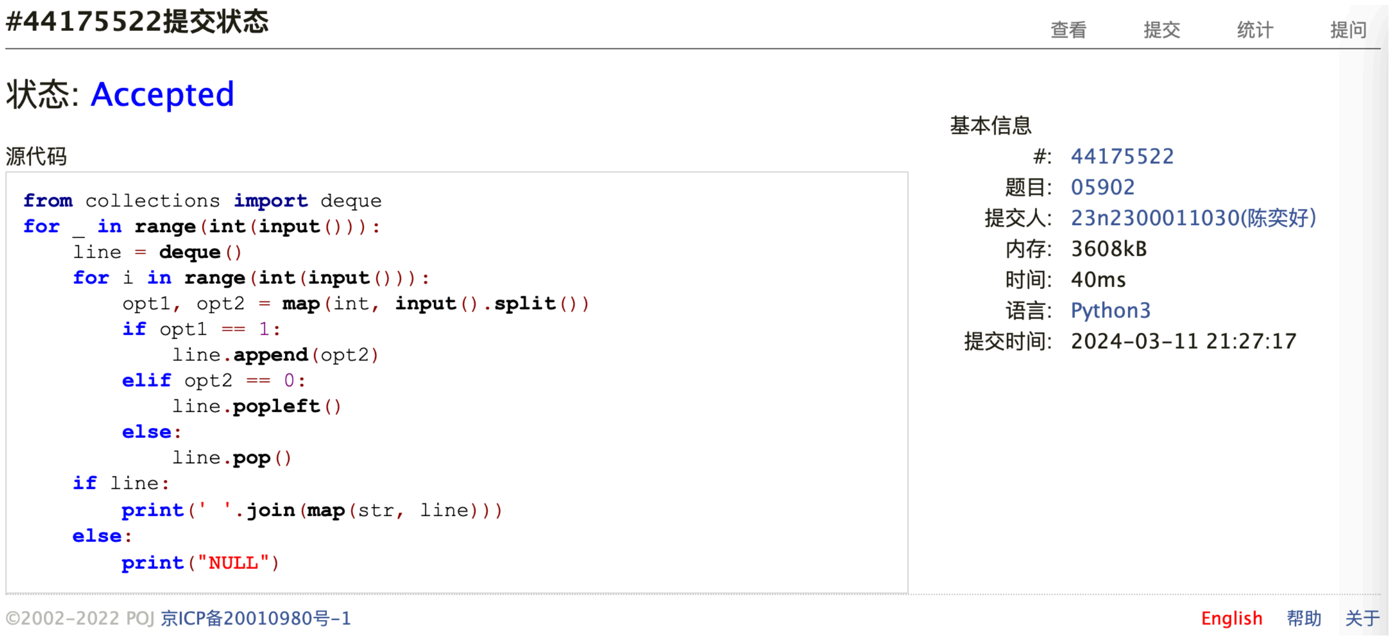
http://cs101.openjudge.cn/practice/05902/

思路：deque是一个双向队列在list操作上还有`appendleft()`,`popleft()`等用法

代码

```
from collections import deque
for _ in range(int(input())):
    line = deque()
    for i in range(int(input())):
        opt1, opt2 = map(int, input().split())
        if opt1 == 1:
            line.append(opt2)
        elif opt2 == 0:
            line.popleft()
        else:
            line.pop()
    if line:
        print(' '.join(map(str, line)))
    else:
        print("NULL")
```

代码运行截图 ==（至少包含有"Accepted"）==


#44175522提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque
for _ in range(int(input())):
    line = deque()
    for i in range(int(input())):
        opt1, opt2 = map(int, input().split())
        if opt1 == 1:
            line.append(opt2)
        elif opt2 == 0:
            line.popleft()
        else:
            line.pop()
    if line:
        print(' '.join(map(str, line)))
    else:
        print("NULL")
```

基本信息

#:	44175522
题目:	05902
提交人:	23n2300011030(陈奕好)
内存:	3608kB
时间:	40ms
语言:	Python3
提交时间:	2024-03-11 21:27:17

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

02694: 波兰表达式

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

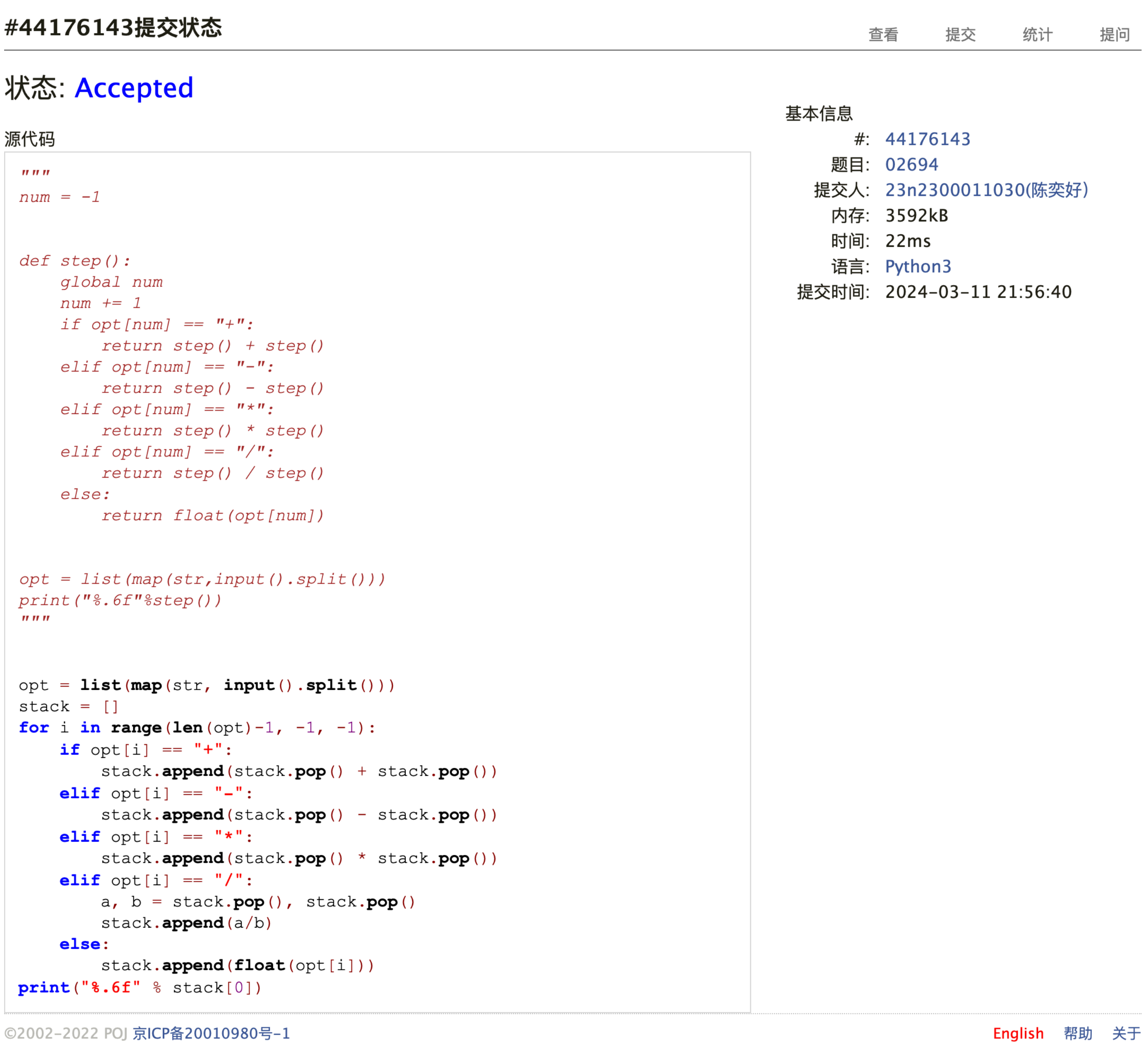
思路：两个方法

代码

```
"""
num = -1


def step():
    global num
```

```python
    num += 1
    if opt[num] == "+":
        return step() + step()
    elif opt[num] == "-":
        return step() - step()
    elif opt[num] == "*":
        return step() * step()
    elif opt[num] == "/":
        return step() / step()
    else:
        return float(opt[num])


opt = list(map(str,input().split()))
print("%.6f"%step())
"""


opt = list(map(str, input().split()))
stack = []
for i in range(len(opt)-1, -1, -1):
    if opt[i] == "+":
        stack.append(stack.pop() + stack.pop())
    elif opt[i] == "-":
        stack.append(stack.pop() - stack.pop())
    elif opt[i] == "*":
        stack.append(stack.pop() * stack.pop())
    elif opt[i] == "/":
        a, b = stack.pop(), stack.pop()
        stack.append(a/b)
    else:
        stack.append(float(opt[i]))
    print(stack)
print("%.6f" % stack[0])
```

代码运行截图 ==（至少包含有"Accepted"）==



#44176143提交状态

状态: Accepted

源代码

```
"""
num = -1


def step():
    global num
    num += 1
    if opt[num] == "+":
        return step() + step()
    elif opt[num] == "-":
        return step() - step()
    elif opt[num] == "*":
        return step() * step()
    elif opt[num] == "/":
        return step() / step()
    else:
        return float(opt[num])


opt = list(map(str,input().split()))
print("%.6f"%step())
"""


opt = list(map(str, input().split()))
stack = []
for i in range(len(opt)-1, -1, -1):
    if opt[i] == "+":
        stack.append(stack.pop() + stack.pop())
    elif opt[i] == "-":
        stack.append(stack.pop() - stack.pop())
    elif opt[i] == "*":
        stack.append(stack.pop() * stack.pop())
    elif opt[i] == "/":
        a, b = stack.pop(), stack.pop()
        stack.append(a/b)
    else:
        stack.append(float(opt[i]))
print("%.6f" % stack[0])
```

基本信息

#:	44176143
题目:	02694
提交人:	23n2300011030(陈奕好)
内存:	3592kB
时间:	22ms
语言:	Python3
提交时间:	2024-03-11 21:56:40

©2002-2022 POJ 京ICP备20010980号-1

24591: 中序表达式转后序表达式

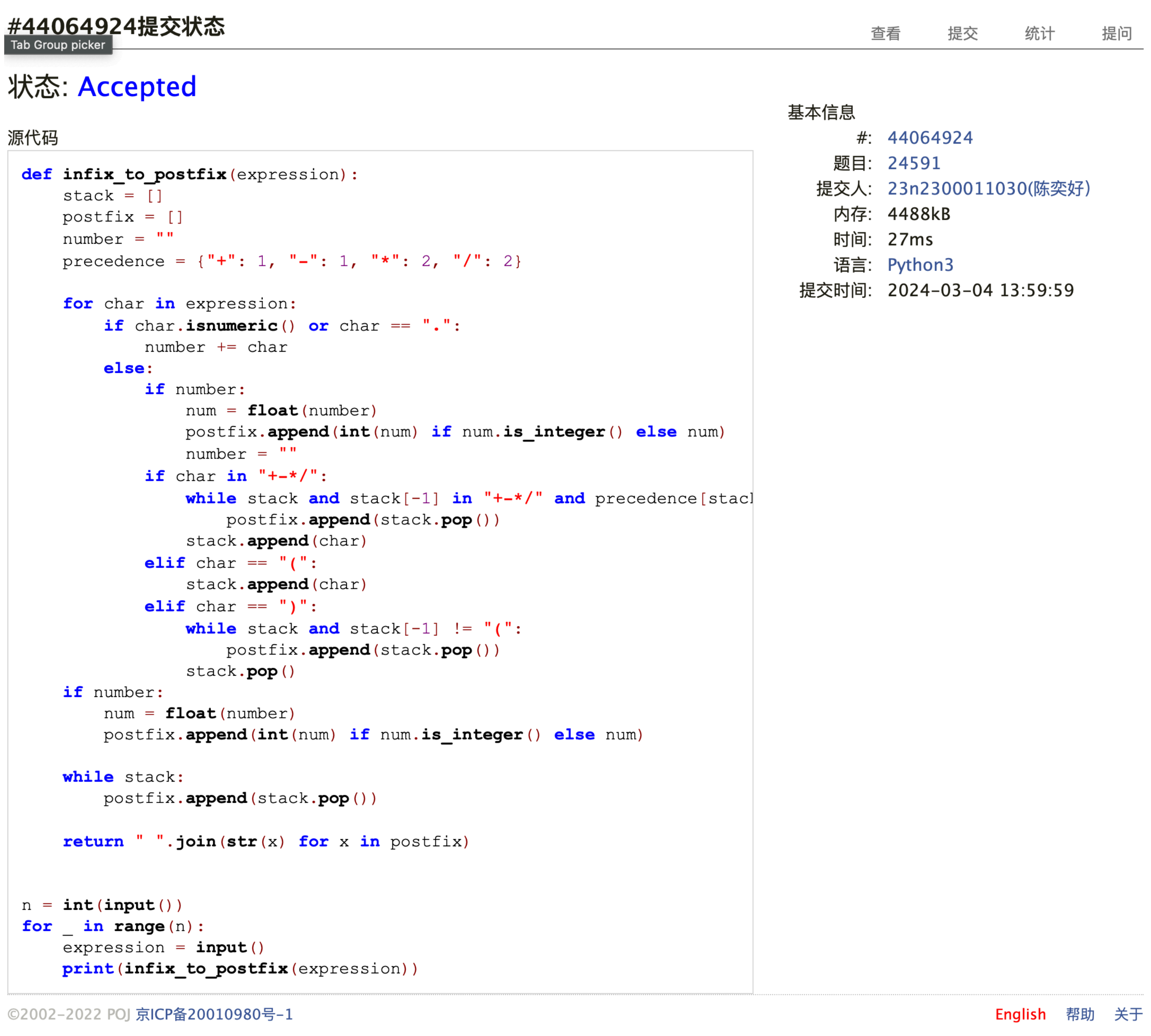
http://cs101.openjudge.cn/practice/24591/

思路：标准

代码

```
def infix_to_postfix(expression):
    stack = []
    postfix = []
    number = ""
    precedence = {"+": 1, "-": 1, "*": 2, "/": 2}
    for char in expression:
        if char.isnumeric() or char == ".":
            number += char
        else:
            if number:
                num = float(number)
                postfix.append(int(num) if num.is_integer() else num)
                number = ""
            if char in "+-*/":
                while stack and stack[-1] in "+-*/" and precedence[stack[-1]] >=
precedence[char]:
                    postfix.append(stack.pop())
                stack.append(char)
            elif char == "(":
                stack.append(char)
            elif char == ")":
                while stack and stack[-1] != "(":
                    postfix.append(stack.pop())
                stack.pop()
    if number:
        num = float(number)
        postfix.append(int(num) if num.is_integer() else num)

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

    return " ".join(str(x) for x in postfix)


n = int(input())
for _ in range(n):
    expression = input()
    print(infix_to_postfix(expression))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44064924提交状态

状态: Accepted

源代码

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

    for char in expression:
        if char.isnumeric() or char == ".":
            number += char
        else:
            if number:
                num = float(number)
                postfix.append(int(num) if num.is_integer() else num)
                number = ""
            if char in "+-*/":
                while stack and stack[-1] in "+-*/" and precedence[stac
                    postfix.append(stack.pop())
                stack.append(char)
            elif char == "(":
                stack.append(char)
            elif char == ")":
                while stack and stack[-1] != "(":
                    postfix.append(stack.pop())
                stack.pop()
    if number:
        num = float(number)
        postfix.append(int(num) if num.is_integer() else num)

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

    return " ".join(str(x) for x in postfix)


n = int(input())
for _ in range(n):
    expression = input()
    print(infix_to_postfix(expression))
```

基本信息

#:	44064924
题目:	24591
提交人:	23n2300011030(陈奕好)
内存:	4488kB
时间:	27ms
语言:	Python3
提交时间:	2024-03-04 13:59:59

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

22068: 合法出栈序列

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

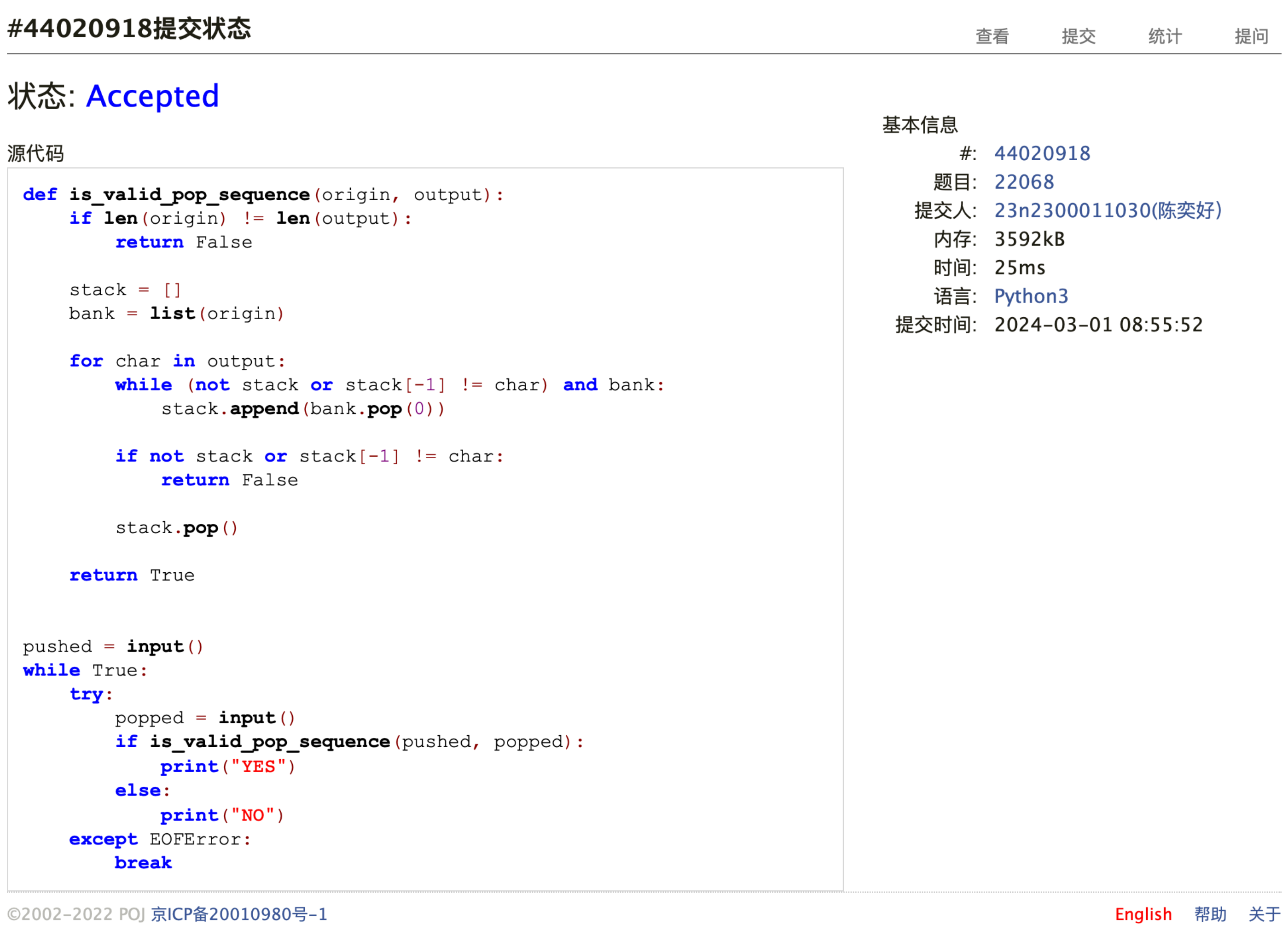
思路：这里开了stack和bank；bank用来存储顺序元素，stack就是栈。元素顺序输出进栈，栈判断能否弹出顶元素，如果不能就存储，否则弹出。直至bank清空，判断stack是否不符合条件（空或顶元素==char）则判断为负

代码

```
def is_valid_pop_sequence(origin, output):
    if len(origin) != len(output):
        return False

    stack = []
```

```python
    bank = list(origin)

    for char in output:
        while (not stack or stack[-1] != char) and bank:
            stack.append(bank.pop(0))

        if not stack or stack[-1] != char:
            return False

        stack.pop()

    return True


pushed = input()
while True:
    try:
        popped = input()
        if is_valid_pop_sequence(pushed, popped):
            print("YES")
        else:
            print("NO")
    except EOFError:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44020918提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```python
def is_valid_pop_sequence(origin, output):
    if len(origin) != len(output):
        return False

    stack = []
    bank = list(origin)

    for char in output:
        while (not stack or stack[-1] != char) and bank:
            stack.append(bank.pop(0))

        if not stack or stack[-1] != char:
            return False

        stack.pop()

    return True


pushed = input()
while True:
    try:
        popped = input()
        if is_valid_pop_sequence(pushed, popped):
            print("YES")
        else:
            print("NO")
    except EOFError:
        break
```

基本信息

#:	44020918
题目:	22068
提交人:	23n2300011030(陈奕好)
内存:	3592kB
时间:	25ms
语言:	Python3
提交时间:	2024-03-01 08:55:52

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

06646: 二叉树的深度

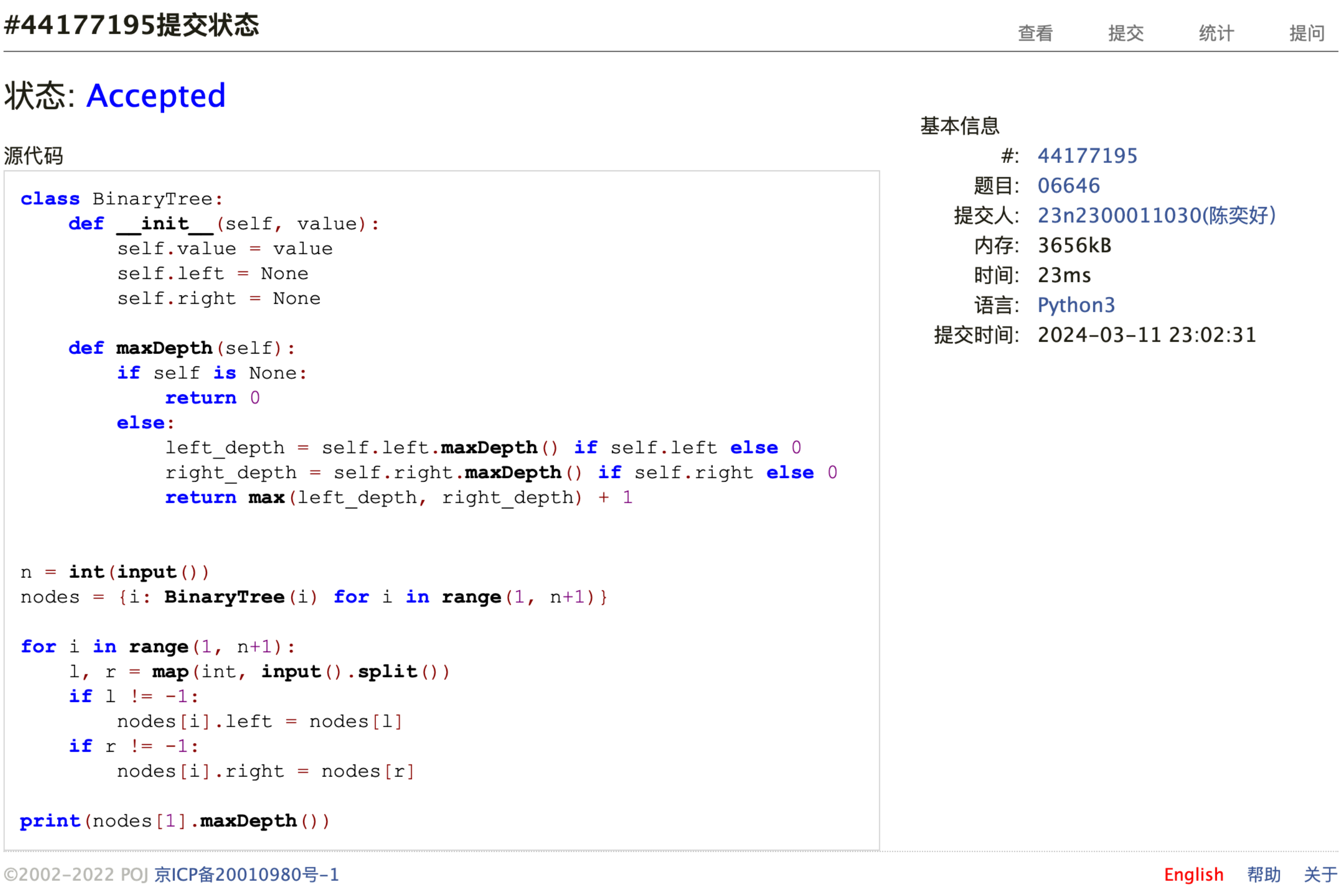
http://cs101.openjudge.cn/practice/06646/

思路：第一次建二叉树，在copilot辅助下进行。

代码

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

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


n = int(input())
nodes = {i: BinaryTree(i) for i in range(1, n+1)}

for i in range(1, n+1):
    l, r = map(int, input().split())
    if l != -1:
        nodes[i].left = nodes[l]
    if r != -1:
        nodes[i].right = nodes[r]

print(nodes[1].maxDepth())
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44177195提交状态

查看 提交 统计 提问

状态: Accepted

源代码

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

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


n = int(input())
nodes = {i: BinaryTree(i) for i in range(1, n+1)}

for i in range(1, n+1):
    l, r = map(int, input().split())
    if l != -1:
        nodes[i].left = nodes[l]
    if r != -1:
        nodes[i].right = nodes[r]

print(nodes[1].maxDepth())
```

基本信息

#:	44177195
题目:	06646
提交人:	23n2300011030(陈奕妤)
内存:	3656kB
时间:	23ms
语言:	Python3
提交时间:	2024-03-11 23:02:31

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

02299: Ultra-QuickSort

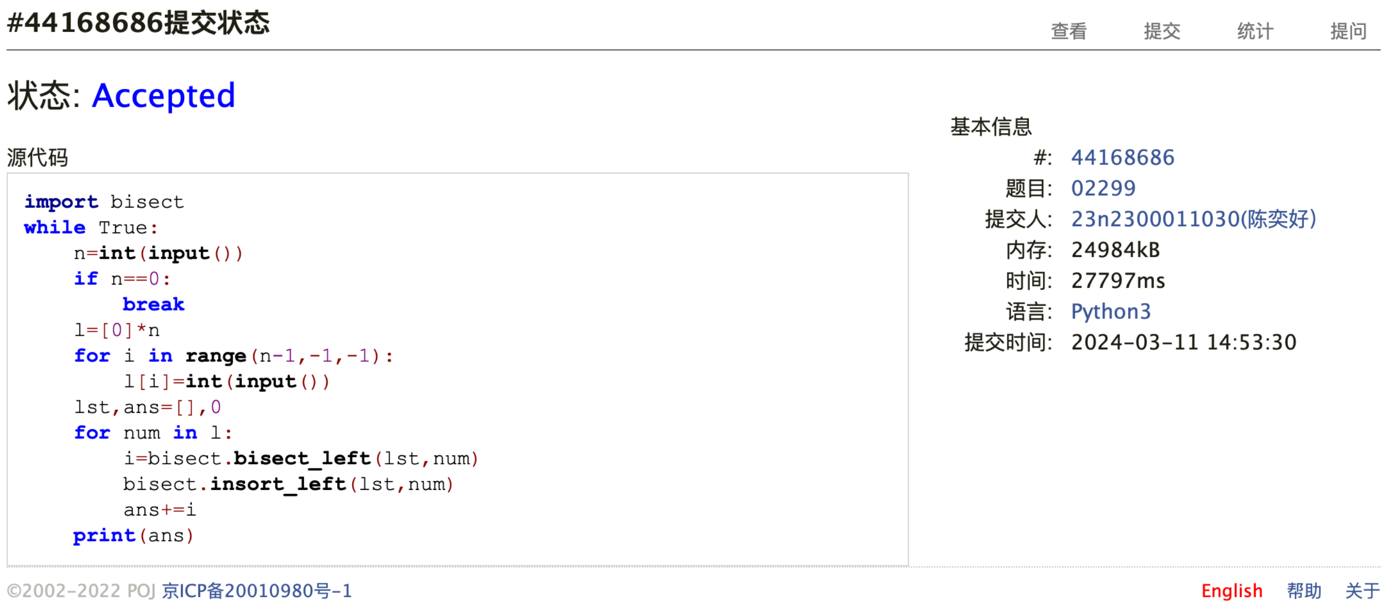
http://cs101.openjudge.cn/practice/02299/

思路：bisect一遍，mergesort一遍

代码

```
import bisect
while True:
    n=int(input())
    if n==0:
        break
    l=[0]*n
    for i in range(n-1,-1,-1):
        l[i]=int(input())
    lst,ans=[],0
    for num in l:
        i=bisect.bisect_left(lst,num)
        bisect.insort_left(lst,num)
        ans+=i
    print(ans)



k = 0
def MergeSort(lists):
    if len(lists) <= 1:
        return lists
    Mid = len(lists)//2
    Left_lists = MergeSort(lists[:Mid])
    Right_lists = MergeSort(lists[Mid:])
    return Merge(Left_lists,Right_lists)

def Merge(Left,Right):
    global k
    Sortedlist = []
    i, j = 0, 0
    while i < len(Left) and j < len(Right):
        # print(i,j)
        if Left[i] <= Right[j]:
            Sortedlist.append(Left[i])
            i += 1
        else:
            Sortedlist.append(Right[j])
            k += len(Left) - i
            j += 1
        # print((Left,Right),k)
    Sortedlist += Left[i:]
    Sortedlist += Right[j:]
    # print(Sortedlist,k)
    return Sortedlist


while True:
    n = int(input())
    if n == 0:
        break
    else:
        k = 0
        arr = [int(input()) for _ in range(n)]
        MergeSort(arr)
        print(k)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==


#44168686提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import bisect
while True:
    n=int(input())
    if n==0:
        break
    l=[0]*n
    for i in range(n-1,-1,-1):
        l[i]=int(input())
    lst,ans=[],0
    for num in l:
        i=bisect.bisect_left(lst,num)
        bisect.insort_left(lst,num)
        ans+=i
    print(ans)
```

基本信息

#:	44168686
题目:	02299
提交人:	23n2300011030(陈奕好)
内存:	24984kB
时间:	27797ms
语言:	Python3
提交时间:	2024-03-11 14:53:30

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

2. 学习总结和收获

==如果作业题目简单，有否额外练习题目，比如：OJ“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。==

其实还是有点题目需要反思，而且有的还是要背的：中序和那道题感觉每次自己打都会漏一点什么，还是需要背诵。