Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Complied by ==黄源森，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：W11

Python编程环境：Spyder IDE 5.2.2

1. 题目

05902: 双端队列

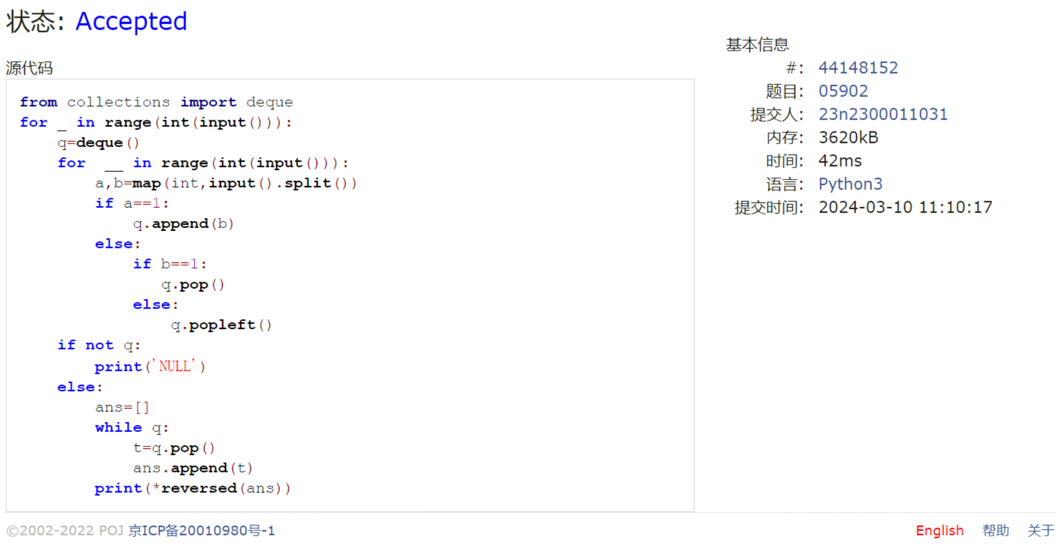
http://cs101.openjudge.cn/practice/05902/

思路：

代码

```
#
from collections import deque
for _ in range(int(input())):
    q=deque()
    for  __ in range(int(input())):
        a,b=map(int,input().split())
        if a==1:
            q.append(b)
        else:
            if b==1:
               q.pop()
            else:
                q.popleft()
    if not q:
        print('NULL')
    else:
        ans=[]
        while q:
            t=q.pop()
            ans.append(t)
        print(*reversed(ans))
```

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

```
from collections import deque
for _ in range(int(input())):
    q=deque()
    for __ in range(int(input())):
        a,b=map(int,input().split())
        if a==1:
            q.append(b)
        else:
            if b==1:
               q.pop()
            else:
                q.popleft()
    if not q:
        print('NULL')
    else:
        ans=[]
        while q:
            t=q.pop()
            ans.append(t)
        print(*reversed(ans))
```

基本信息

#:	44148152
题目:	05902
提交人:	23n2300011031
内存:	3620kB
时间:	42ms
语言:	Python3
提交时间:	2024-03-10 11:10:17

©2002-2022 POJ 京ICP备20010980号-1

02694: 波兰表达式

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

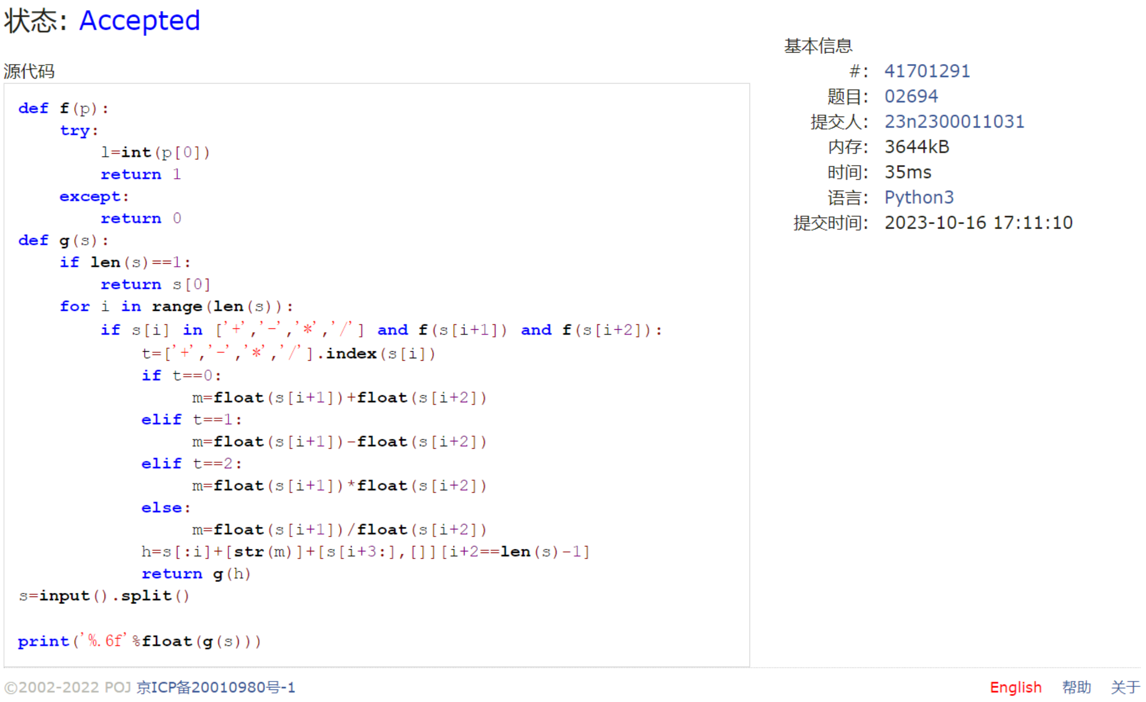
思路:

代码

```
#
def f(p):
    try:
        l=int(p[0])
        return 1
    except:
        return 0
def g(s):
    if len(s)==1:
        return s[0]
    for i in range(len(s)):
        if s[i] in ['+','-','*','/'] and f(s[i+1]) and f(s[i+2]):
            t=['+','-','*','/'].index(s[i])
            if t==0:
                 m=float(s[i+1])+float(s[i+2])
            elif t==1:
                 m=float(s[i+1])-float(s[i+2])
            elif t==2:
                 m=float(s[i+1])*float(s[i+2])
            else:
                 m=float(s[i+1])/float(s[i+2])
            h=s[:i]+[str(m)]+[s[i+3:],[]][i+2==len(s)-1]
            return g(h)
s=input().split()

print('%.6f'%float(g(s)))
```

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

```
def f(p):
    try:
        l=int(p[0])
        return 1
    except:
        return 0
def g(s):
    if len(s)==1:
        return s[0]
    for i in range(len(s)):
        if s[i] in ['+','-','*','/'] and f(s[i+1]) and f(s[i+2]):
            t=['+','-','*','/'].index(s[i])
            if t==0:
                 m=float(s[i+1])+float(s[i+2])
            elif t==1:
                 m=float(s[i+1])-float(s[i+2])
            elif t==2:
                 m=float(s[i+1])*float(s[i+2])
            else:
                 m=float(s[i+1])/float(s[i+2])
            h=s[:i]+[str(m)]+[s[i+3:],[]][i+2==len(s)-1]
            return g(h)
s=input().split()

print('%.6f'%float(g(s)))
```

基本信息

#:	41701291
题目:	02694
提交人:	23n2300011031
内存:	3644kB
时间:	35ms
语言:	Python3
提交时间:	2023-10-16 17:11:10

©2002-2022 POJ 京ICP备20010980号-1

24591: 中序表达式转后序表达式

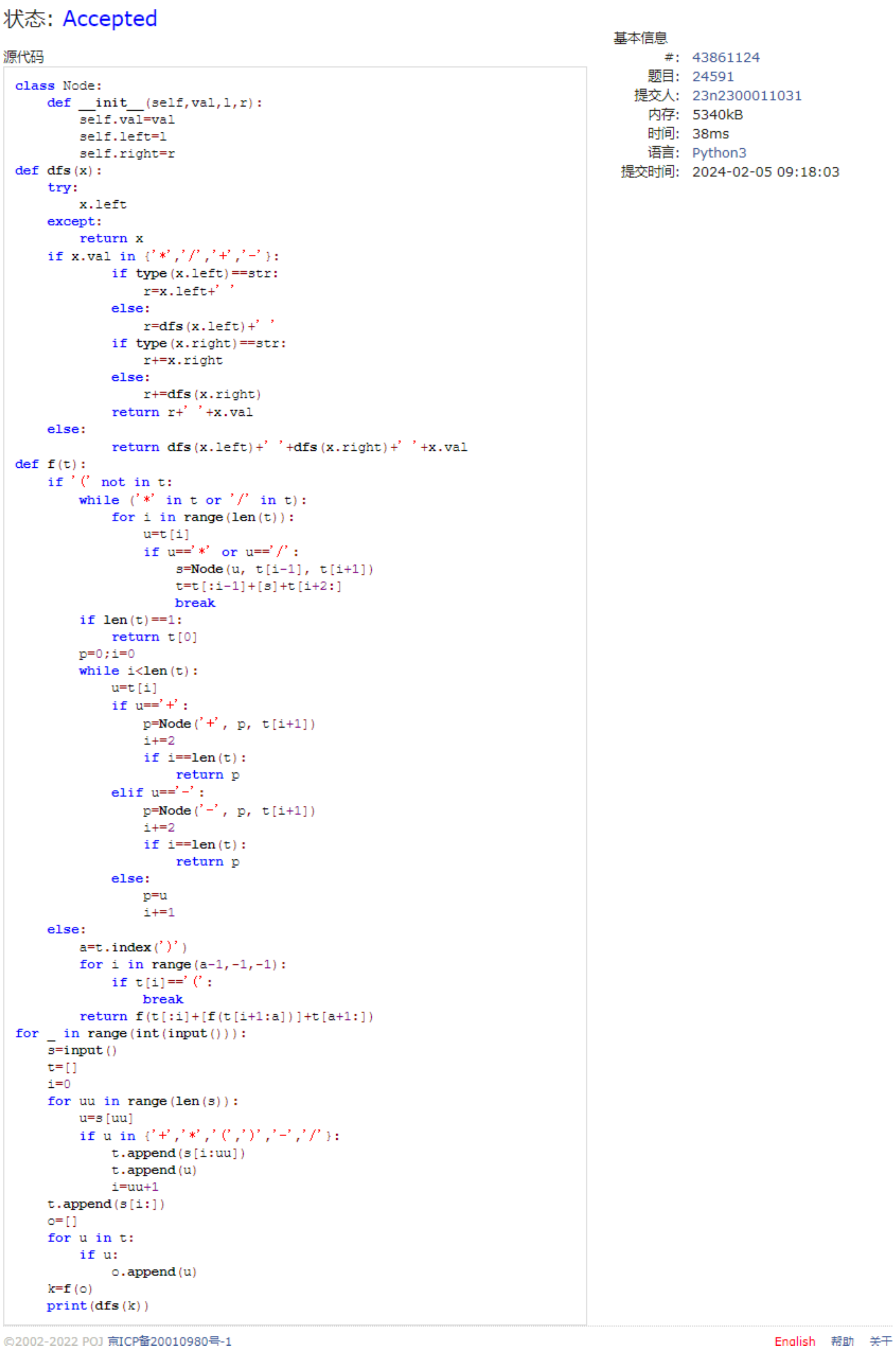
http://cs101.openjudge.cn/practice/24591/

思路：

代码

```
#
class Node:
    def __init__(self,val,l,r):
        self.val=val
        self.left=l
        self.right=r
def dfs(x):
    try:
        x.left
    except:
        return x
    if x.val in {'*','/','+','-'}:
            if type(x.left)==str:
                r=x.left+' '
            else:
                r=dfs(x.left)+' '
            if type(x.right)==str:
                r+=x.right
            else:
                r+=dfs(x.right)
            return r+' '+x.val
    else:
            return dfs(x.left)+' '+dfs(x.right)+' '+x.val
def f(t):
    if '(' not in t:
        while ('*' in t or '/' in t):
            for i in range(len(t)):
                u=t[i]
                if u=='*' or u=='/':
                    s=Node(u, t[i-1], t[i+1])
                    t=t[:i-1]+[s]+t[i+2:]
                    break
        if len(t)==1:
            return t[0]
        p=0;i=0
        while i<len(t):
            u=t[i]
            if u=='+':
                p=Node('+', p, t[i+1])
                i+=2
                if i==len(t):
                    return p
            elif u=='-':
                p=Node('-', p, t[i+1])
                i+=2
                if i==len(t):
                    return p
            else:
                p=u
                i+=1
    else:
        a=t.index(')')
        for i in range(a-1,-1,-1):
            if t[i]=='(':
                break
        return f(t[:i]+[f(t[i+1:a])]+t[a+1:])
for _ in range(int(input())):
    s=input()
    t=[]
    i=0
    for uu in range(len(s)):
        u=s[uu]
        if u in {'+','*','(',')','-','/'}:
            t.append(s[i:uu])
            t.append(u)
            i=uu+1
    t.append(s[i:])
    o=[]
    for u in t:
        if u:
            o.append(u)
    k=f(o)
    print(dfs(k))
```

代码运行截图 == （AC代码截图，至少包含有"Accepted"） ==



状态: Accepted

源代码

```python
class Node:
    def __init__(self,val,l,r):
        self.val=val
        self.left=l
        self.right=r
def dfs(x):
    try:
        x.left
    except:
        return x
    if x.val in {'*','/','+','-'}:
            if type(x.left)==str:
                r=x.left+' '
            else:
                r=dfs(x.left)+' '
            if type(x.right)==str:
                r+=x.right
            else:
                r+=dfs(x.right)
            return r+' '+x.val
    else:
            return dfs(x.left)+' '+dfs(x.right)+' '+x.val
def f(t):
    if '(' not in t:
        while ('*' in t or '/' in t):
            for i in range(len(t)):
                u=t[i]
                if u=='*' or u=='/':
                    s=Node(u, t[i-1], t[i+1])
                    t=t[:i-1]+[s]+t[i+2:]
                    break
        if len(t)==1:
            return t[0]
        p=0;i=0
        while i<len(t):
            u=t[i]
            if u=='+':
                p=Node('+', p, t[i+1])
                i+=2
                if i==len(t):
                    return p
            elif u=='-':
                p=Node('-', p, t[i+1])
                i+=2
                if i==len(t):
                    return p
            else:
                p=u
                i+=1
    else:
        a=t.index(')')
        for i in range(a-1,-1,-1):
            if t[i]=='(':
                break
        return f(t[:i]+[f(t[i+1:a])]+t[a+1:])
for _ in range(int(input())):
    s=input()
    t=[]
    i=0
    for uu in range(len(s)):
        u=s[uu]
        if u in {'+','*','(',')','-','/'}:
            t.append(s[i:uu])
            t.append(u)
            i=uu+1
    t.append(s[i:])
    o=[]
    for u in t:
        if u:
            o.append(u)
    k=f(o)
    print(dfs(k))
```

基本信息

#:	43861124
题目:	24591
提交人:	23n2300011031
内存:	5340kB
时间:	38ms
语言:	Python3
提交时间:	2024-02-05 09:18:03

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

22068: 合法出栈序列

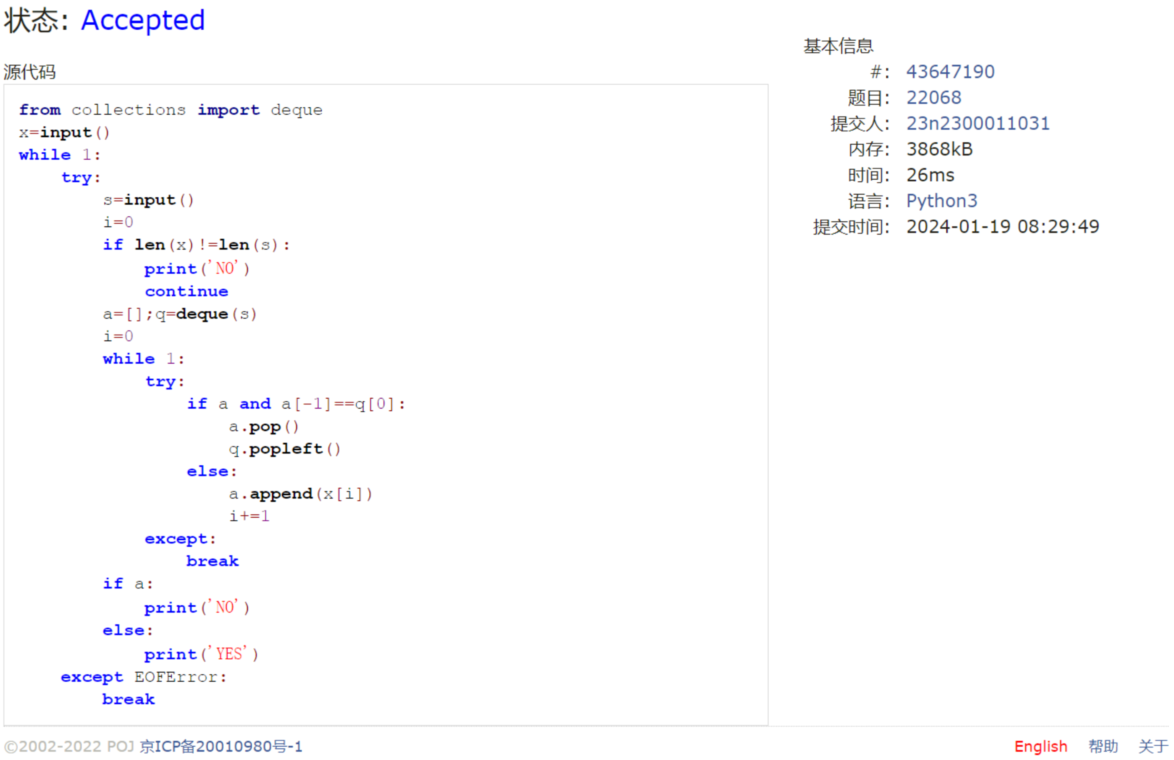
http://cs101.openjudge.cn/practice/22068/

思路:

代码

```
#
from collections import deque
x=input()
while 1:
    try:
        s=input()
        i=0
        if len(x)!=len(s):
            print('NO')
            continue
        a=[];q=deque(s)
        i=0
        while 1:
            try:
                if a and a[-1]==q[0]:
                    a.pop()
                    q.popleft()
                else:
                    a.append(x[i])
                    i+=1
            except:
                break
        if a:
            print('NO')
        else:
            print('YES')
    except EOFError:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```
from collections import deque
x=input()
while 1:
    try:
        s=input()
        i=0
        if len(x)!=len(s):
            print('NO')
            continue
        a=[];q=deque(s)
        i=0
        while 1:
            try:
                if a and a[-1]==q[0]:
                    a.pop()
                    q.popleft()
                else:
                    a.append(x[i])
                    i+=1
            except:
                break
        if a:
            print('NO')
        else:
            print('YES')
    except EOFError:
        break
```

基本信息

#:	43647190
题目:	22068
提交人:	23n2300011031
内存:	3868kB
时间:	26ms
语言:	Python3
提交时间:	2024-01-19 08:29:49

06646: 二叉树的深度

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

思路:

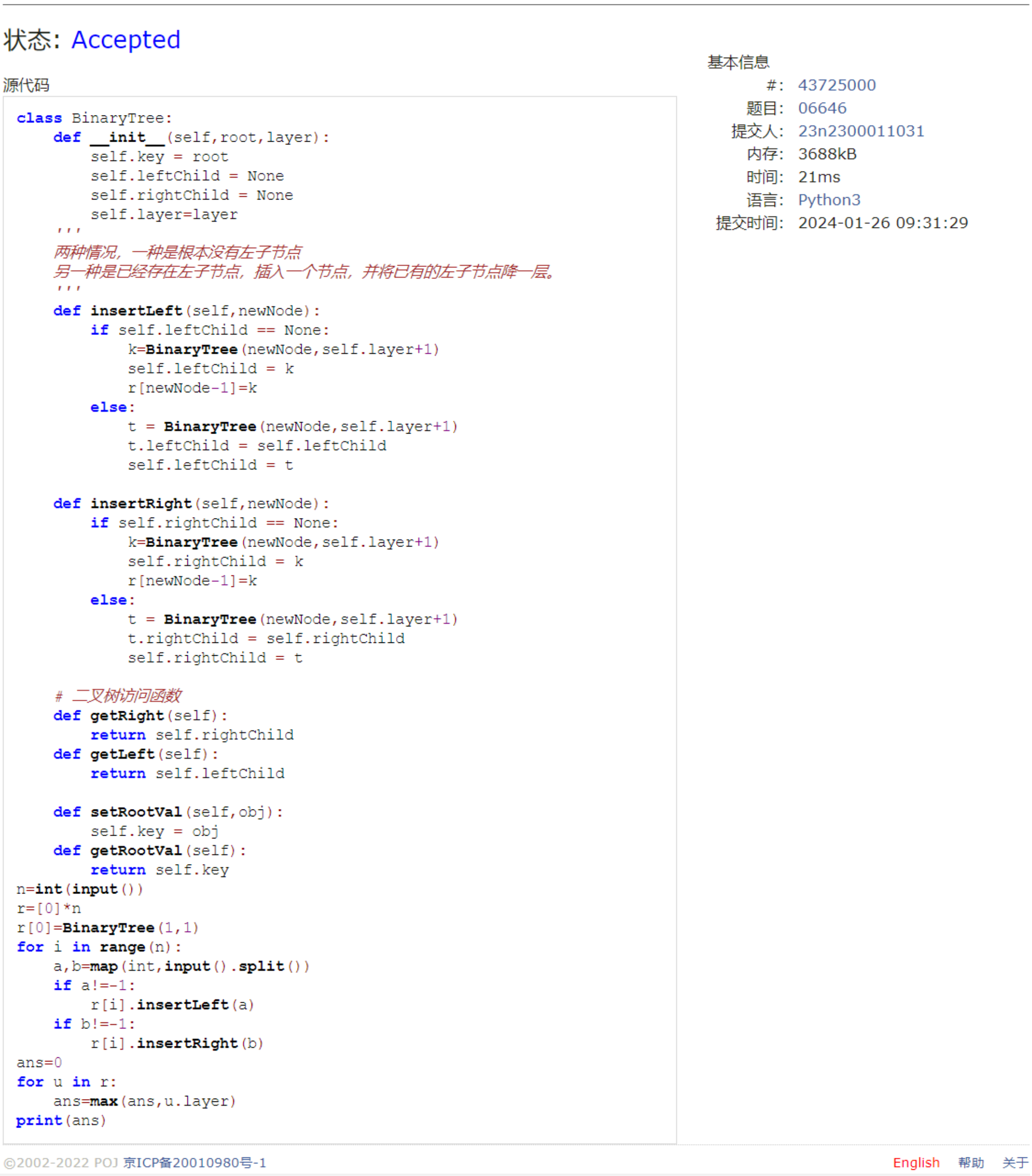
代码

```
#
class BinaryTree:
    def __init__(self,root,layer):
        self.key = root
        self.leftChild = None
        self.rightChild = None
        self.layer=layer
    '''
    两种情况，一种是根本没有左子节点
    另一种是已经存在左子节点，插入一个节点，并将已有的左子节点降一层。
    '''
    def insertLeft(self,newNode):
        if self.leftChild == None:
            k=BinaryTree(newNode,self.layer+1)
            self.leftChild = k
            r[newNode-1]=k
        else:
            t = BinaryTree(newNode,self.layer+1)
            t.leftChild = self.leftChild
            self.leftChild = t
    def insertRight(self,newNode):
        if self.rightChild == None:
            k=BinaryTree(newNode,self.layer+1)
            self.rightChild = k
            r[newNode-1]=k
        else:
            t = BinaryTree(newNode,self.layer+1)
            t.rightChild = self.rightChild
            self.rightChild = t

    # 二叉树访问函数
    def getRight(self):
        return self.rightChild
    def getLeft(self):
        return self.leftChild

    def setRootVal(self,obj):
        self.key = obj
    def getRootVal(self):
        return self.key
n=int(input())
r=[0]*n
r[0]=BinaryTree(1,1)
for i in range(n):
    a,b=map(int,input().split())
    if a!=-1:
        r[i].insertLeft(a)
    if b!=-1:
        r[i].insertRight(b)
ans=0
for u in r:
    ans=max(ans,u.layer)
print(ans)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```
class BinaryTree:
    def __init__(self,root,layer):
        self.key = root
        self.leftChild = None
        self.rightChild = None
        self.layer=layer
    '''
    两种情况，一种是根本没有左子节点
    另一种是已经存在左子节点，插入一个节点，并将已有的左子节点降一层。
    '''
    def insertLeft(self,newNode):
        if self.leftChild == None:
            k=BinaryTree(newNode,self.layer+1)
            self.leftChild = k
            r[newNode-1]=k
        else:
            t = BinaryTree(newNode,self.layer+1)
            t.leftChild = self.leftChild
            self.leftChild = t

    def insertRight(self,newNode):
        if self.rightChild == None:
            k=BinaryTree(newNode,self.layer+1)
            self.rightChild = k
            r[newNode-1]=k
        else:
            t = BinaryTree(newNode,self.layer+1)
            t.rightChild = self.rightChild
            self.rightChild = t

    # 二叉树访问函数
    def getRight(self):
        return self.rightChild
    def getLeft(self):
        return self.leftChild

    def setRootVal(self,obj):
        self.key = obj
    def getRootVal(self):
        return self.key
n=int(input())
r=[0]*n
r[0]=BinaryTree(1,1)
for i in range(n):
    a,b=map(int,input().split())
    if a!=-1:
        r[i].insertLeft(a)
    if b!=-1:
        r[i].insertRight(b)
ans=0
for u in r:
    ans=max(ans,u.layer)
print(ans)
```

基本信息

#:	43725000
题目:	06646
提交人:	23n2300011031
内存:	3688kB
时间:	21ms
语言:	Python3
提交时间:	2024-01-26 09:31:29

©2002-2022 POJ 京ICP备20010980号-1

02299: Ultra-QuickSort

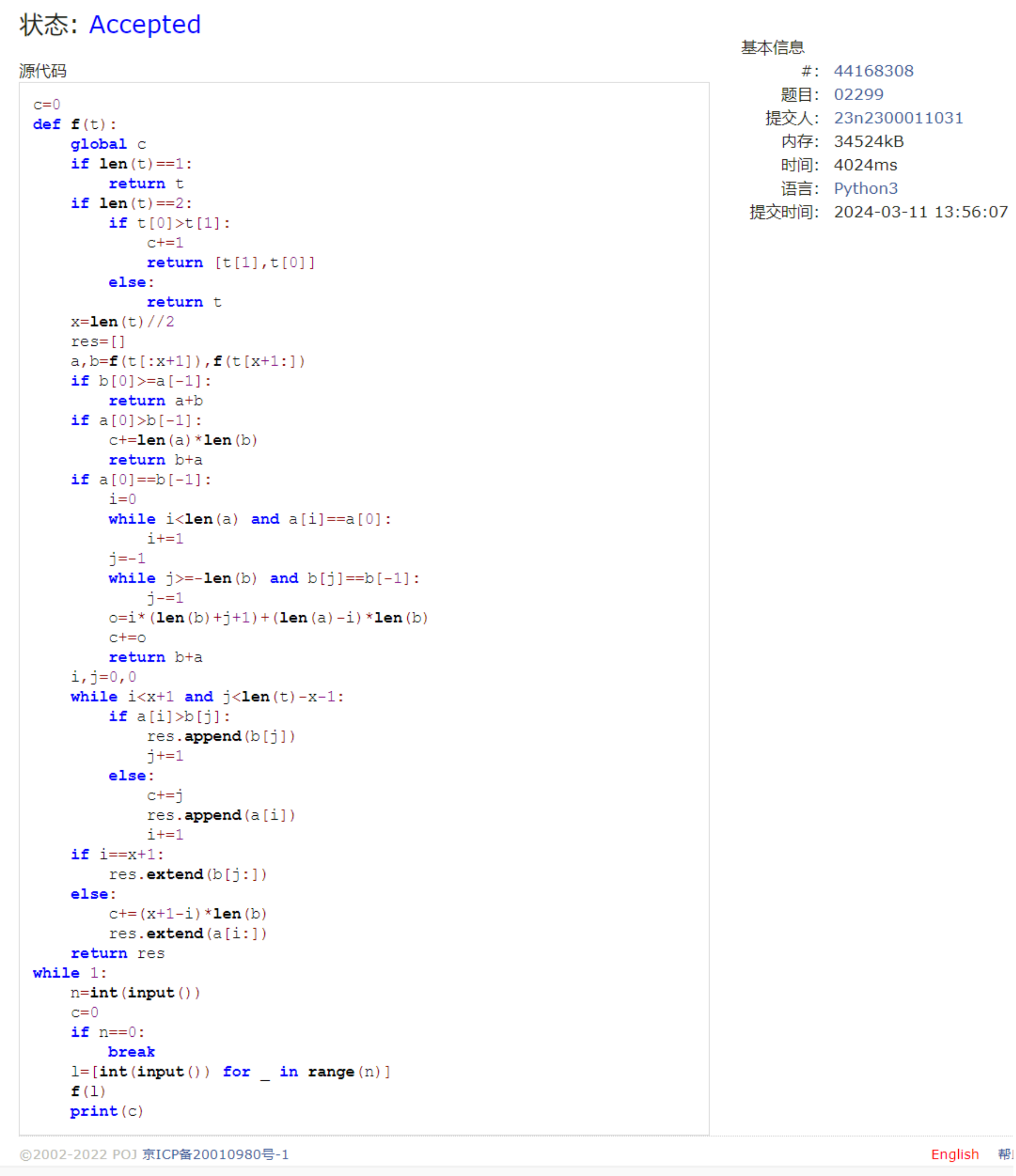
http://cs101.openjudge.cn/practice/02299/

思路：

代码

```
#
c=0
def f(t):
    global c
    if len(t)==1:
        return t
    if len(t)==2:
        if t[0]>t[1]:
            c+=1
            return [t[1],t[0]]
        else:
            return t
    x=len(t)//2
    res=[]
    a,b=f(t[:x+1]),f(t[x+1:])
    if b[0]>=a[-1]:
        return a+b
    if a[0]>b[-1]:
        c+=len(a)*len(b)
        return b+a
    if a[0]==b[-1]:
        i=0
        while i<len(a) and a[i]==a[0]:
            i+=1
        j=-1
        while j>=-len(b) and b[j]==b[-1]:
            j-=1
        o=i*(len(b)+j+1)+(len(a)-i)*len(b)
        c+=o
        return b+a
    i,j=0,0
    while i<x+1 and j<len(t)-x-1:
        if a[i]>b[j]:
            res.append(b[j])
            j+=1
        else:
            c+=j
            res.append(a[i])
            i+=1
    if i==x+1:
        res.extend(b[j:])
    else:
        c+=(x+1-i)*len(b)
        res.extend(a[i:])
    return res
while 1:
    n=int(input())
    c=0
    if n==0:
        break
    l=[int(input()) for _ in range(n)]
    f(l)
    print(c)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```
c=0
def f(t):
    global c
    if len(t)==1:
        return t
    if len(t)==2:
        if t[0]>t[1]:
            c+=1
            return [t[1],t[0]]
        else:
            return t
    x=len(t)//2
    res=[]
    a,b=f(t[:x+1]),f(t[x+1:])
    if b[0]>=a[-1]:
        return a+b
    if a[0]>b[-1]:
        c+=len(a)*len(b)
        return b+a
    if a[0]==b[-1]:
        i=0
        while i<len(a) and a[i]==a[0]:
            i+=1
        j=-1
        while j>=-len(b) and b[j]==b[-1]:
            j-=1
        o=i*(len(b)+j+1)+(len(a)-i)*len(b)
        c+=o
        return b+a
    i,j=0,0
    while i<x+1 and j<len(t)-x-1:
        if a[i]>b[j]:
            res.append(b[j])
            j+=1
        else:
            c+=j
            res.append(a[i])
            i+=1
    if i==x+1:
        res.extend(b[j:])
    else:
        c+=(x+1-i)*len(b)
        res.extend(a[i:])
    return res
while 1:
    n=int(input())
    c=0
    if n==0:
        break
    l=[int(input()) for _ in range(n)]
    f(l)
    print(c)
```

基本信息

#:	44168308
题目:	02299
提交人:	23n2300011031
内存:	34524kB
时间:	4024ms
语言:	Python3
提交时间:	2024-03-11 13:56:07

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

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。==

有些题目是寒假用树做的，现在尝试将树和递归、栈这些联系起来，写更为简洁的代码，也做些其他website的题