

Assignment #6: "树"算: Huffman,BinHeap,BST,AVL,DisjointSet

Updated 2214 GMT+8 March 24, 2024

2024 spring, Compiled by 郑铭毅 数学科学学院

说明:

- 1) 这次作业内容不简单, 耗时长直接参考题解。
- 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

Windows 11

PyCharm

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

22275: 二叉搜索树的遍历

<http://cs101.openjudge.cn/practice/22275/>

思路:

代码

```

class Treenode:
    def __init__(self,value):
        self.value=value
        self.leftson=None
        self.rightson=None
    def output(self):
        l=[]
        if self.leftson:
            l.extend(self.leftson.output())
        if self.rightson:
            l.extend(self.rightson.output())
        l.append(self.value)
        return l
def insert_into_bst(root, value):
    if root==None:
        return Treenode(value)
    else:
        if root.value<value:
            root.rightson=insert_into_bst(root.rightson,value)
        elif root.value>value:
            root.leftson=insert_into_bst(root.leftson,value)
        return root
def build_tree(s):
    root=None
    for i in s:
        root=insert_into_bst(root,i)
    return root
n=int(input())
s=[int(i) for i in input().split()]
l=build_tree(s).output()
for i in range(n-1):
    print(int(l[i]),end=' ')
print(int(l[-1]))

```


OpenJudge

题目ID, 标题, 描述

2300010872

信箱

账号

 **CS101 / 题库**

[题目](#) [排名](#) [状态](#) [提问](#)

#44445892提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
class Treenode:
    def __init__(self,value):
        self.value=value
        self.leftson=None
        self.rightson=None
    def output(self):
        l=[]
        if self.leftson:
            l.extend(self.leftson.output())
        if self.rightson:
            l.extend(self.rightson.output())
        l.append(self.value)
        return l
    def insert_into_bst(root, value):
        if root==None:
            return Treenode(value)
        else:
            if root.value<value:
                root.rightson=insert_into_bst(root.rightson,value)
            elif root.value>value:
                root.leftson=insert_into_bst(root.leftson,value)
            return root
    def build_tree(s):
        root=None
        for i in s:
            root=insert_into_bst(root,i)
        return root
n=int(input())
s=[int(i) for i in input().split()]
l=build_tree(s).output()
for i in range(n-1):
    print(int(l[i]),end=' ')
print(int(l[-1]))
```

基本信息

#: 44445892

题目: 22275

提交人: 2300010872

内存: 4004kB

时间: 29ms

语言: Python3

提交时间: 2024-03-29 16:20:16

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

05455: 二叉搜索树的层次遍历

<http://cs101.openjudge.cn/practice/05455/>

思路:

代码

```

class Treenode:

    def __init__(self,value):

        self.value=value

        self.leftson=None

        self.rightson=None

    def output(self):

        l=[]

        result=[]

        l.append(self)

        while l:

            s=[]

            for i in range(len(l)):

                if l[i].leftson:

                    s.append(l[i].leftson)

                if l[i].rightson:

                    s.append(l[i].rightson)

                result.append(l[i].value)

            l=s

        return result

def insert_into_bst(root, value):

    if root==None:

        return Treenode(value)

    else:

        if root.value<value:

            root.rightson=insert_into_bst(root.rightson,value)

        elif root.value>value:

            root.leftson=insert_into_bst(root.leftson,value)

        return root

def build_tree(s):

    root=None

    for i in s:

        root=insert_into_bst(root,i)

    return root

s=[int(i) for i in input().split()]

L=[]

for i in s:

    if i not in L:

        L.append(i)

l=build_tree(L).output()

for i in range(len(l)-1):

    print(l[i],end=' ')

print(l[-1])

```

状态: Accepted

源代码

```
class Treenode:
    def __init__(self, value):
        self.value = value
        self.leftson = None
        self.rightson = None
    def output(self):
        l = []
        result = []
        l.append(self)
        while l:
            s = []
            for i in range(len(l)):
                if l[i].leftson:
                    s.append(l[i].leftson)
                if l[i].rightson:
                    s.append(l[i].rightson)
                result.append(l[i].value)
            l = s
        return result

def insert_into_bst(root, value):
    if root == None:
        return Treenode(value)
    else:
        if root.value < value:
            root.rightson = insert_into_bst(root.rightson, value)
        elif root.value > value:
            root.leftson = insert_into_bst(root.leftson, value)
        return root

def build_tree(s):
    root = None
    for i in s:
        root = insert_into_bst(root, i)
    return root

s = [int(i) for i in input().split()]
L = []
for i in s:
    if i not in L:
        L.append(i)
l = build_tree(L).output()
for i in range(len(l)-1):
    print(l[i], end=' ')
print(l[-1])
```

基本信息

#: 44498819
题目: 05455
提交人: 2300010872
内存: 5120kB
时间: 281ms
语言: Python3
提交时间: 2024-04-01 19:32:44

04078: 实现堆结构

<http://cs101.openjudge.cn/practice/04078/>

练习自己写个BinHeap。当然机考时候，如果遇到这样题目，直接import heapq。手搓栈、队列、堆、AVL等，考试前需要搓个遍。

思路:

代码

```

def f(l,a):
    if l:
        if a>=l[0] and a<l[-1]:

            for i in range(len(l)-1):
                if a>=l[i] and a<l[i+1]:
                    l.insert(i+1,a)
                    break

            elif a<l[0]:
                l.insert(0,a)

            elif a>=l[-1]:
                l.append(a)

        else:
            l.append(a)

arr=[]
n=int(input())
for i in range(n):
    s=input().split()

    if s[0]=='2':
        print(arr[0])
        arr.pop(0)

    elif s[0]=='1':
        f(arr,int(s[1]))

```

代码运行截图

OpenJudge
题目ID, 标题, 描述
2300010872
信息
账号

CS101 / 题库
题目 排名 状态 提问

#44499354提交状态
查看 提交 统计 提问

状态: Accepted

源代码

```

def f(l,a):
    if l:
        if a>=l[0] and a<l[-1]:
            for i in range(len(l)-1):
                if a>=l[i] and a<l[i+1]:
                    l.insert(i+1,a)
                    break
            elif a<l[0]:
                l.insert(0,a)
            elif a>=l[-1]:
                l.append(a)
        else:
            l.append(a)

arr=[]
n=int(input())
for i in range(n):
    s=input().split()
    if s[0]=='2':
        print(arr[0])
        arr.pop(0)
    elif s[0]=='1':
        f(arr,int(s[1]))

```

基本信息

44499354
题目: 04078
提交人: 2300010872
内存: 4612kB
时间: 336ms
语言: Python3
提交时间: 2024-04-01 20:08:08

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

22161: 哈夫曼编码树

<http://cs101.openjudge.cn/practice/22161/>

思路:

代码

```
1 #  
2
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

晴问9.5: 平衡二叉树的建立

<https://sunnywhy.com/sfbj/9/5/359>

思路:

代码

```
1 #  
2
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

02524: 宗教信仰

<http://cs101.openjudge.cn/practice/02524/>

思路:

代码

```
def get_father(x,f):
    if f[x]!=x:
        f[x]=get_father(f[x],f)
    return f[x]
def join(x,y,f):
    fx=get_father(x,f)
    fy=get_father(y,f)
    if fx==fy:
        return
    f[fx]=fy
case_num=0
while True:
    n,m=map(int,input().split())
    if n==0 and m==0:
        break
    else:
        f=list(range(n))
        for _ in range(m):
            a,b=map(int,input().split())
            join(a-1,b-1,f)

        count=0
        for i in range(n):
            if f[i]==i:
                count+=1
        case_num+=1
        print(f"Case {case_num}: {count}")
```

代码运行截图



2. 学习总结和收获

本周题目难度确实大，耗费的时间很多，当然收获也很大。感觉对于二叉树相关的遍历问题已经有一定的熟练度，但是如果题目进行一些包装，可能就不太好看出来需要利用树的相关内容来解题，这一块还需加强练习。