# Assignment #9: Huffman, BST & Heap

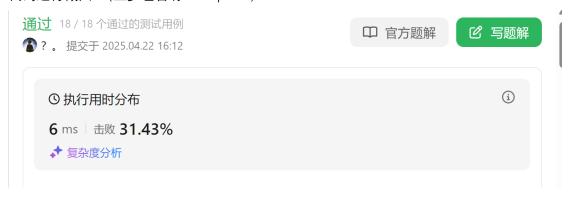
Updated 1834 GMT+8 Apr 15, 2025 2025 spring, Complied by 同学的姓名、院系

#### 1. 题目

#### LC222.完全二叉树的节点个数

```
dfs, <a href="https://leetcode.cn/problems/count-complete-tree-nodes/代码:">https://leetcode.cn/problems/count-complete-tree-nodes/代码:</a>
class Solution:
    def countNodes(self, root: Optional[TreeNode]) -> int:
        if not root:
        return 0
        left = self.countNodes(root.left)
        right = self.countNodes(root.right)
        return left + right + 1
```

# 代码运行截图 (至少包含有"Accepted")



## LC103.二叉树的锯齿形层序遍历

```
bfs, <a href="https://leetcode.cn/problems/binary-tree-zigzag-level-order-traversal/">https://leetcode.cn/problems/binary-tree-zigzag-level-order-traversal/</a>
代码:
from collections import deque
class Solution:
     def zigzagLevelOrder(self, root: Optional[TreeNode]) -> List[List[int]]:
           if not root:
                 return []
           result = []
           q = deque([root])
           while q:
                 tmp = deque()
                 for i in range(len(q)):
                       node = q.popleft()
                       tmp.append(node.val) if len(result) % 2 == 0 else tmp.insert(0,node.val)
                       if node.left:
                             q.append(node.left)
                       if node.right:
                             q.append(node.right)
                 result.append(list(tmp))
           return result
```

# 代码运行截图 (至少包含有"Accepted")



#### M04080:Huffman 编码树

greedy, http://cs101.openjudge.cn/practice/04080/

思路:参考了题解,学习了这个用堆的办法。相比直接建树,减少了算深度的麻烦,不过 挺难想的对我来说。

```
| 代码:
| import heapq | n = int(input()) | quan = list(map(int, input().split())) | heapq.heapify(quan) | total = 0 |
| while len(quan) > 1: | a = heapq.heappop(quan) | b = heapq.heappop(quan) | s = a + b | total += s | heapq.heappush(quan, s)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

print(total)

```
import heapq
n = int(input())
quan = list(map(int, input().split()))
heapq.heapify(quan)
total = 0
while len(quan) > 1:
    a = heapq.heappop(quan)
    b = heapq.heappop(quan)
    s = a + b
    total += s
    heapq.heappush(quan, s)
print(total)
```

基本信息

#: 48987280 题目: 04080 提交人: 2400011037 内存: 3612kB 时间: 20ms 语言: Python3

提交时间: 2025-04-22 21:19:04

```
M05455: 二叉搜索树的层次遍历
http://cs101.openjudge.cn/practice/05455/
代码:
class TreeNode:
    def __init__(self, value):
         self.value = value
         self.left = None
         self.right = None
from collections import deque
def insert(root,value):
    if root == None:
         return TreeNode(value)
    if root.value == value:
         return root
    elif root.value > value:
         root.left = insert(root.left,value)
    else:
         root.right = insert(root.right,value)
    return root
lst = list(map(int, input().split()))
root = None
for i in lst:
    root = insert(root,i)
res = []
q = deque([root])
while q:
    node = q.popleft()
    res.append(node.value)
    if node.left:
         q.append(node.left)
    if node.right:
         q.append(node.right)
print(" ".join(map(str, res)))
代码运行截图 (至少包含有"Accepted")
  状态: Accepted
                                                                  基本信息
  源代码
                                                                       #: 48987651
                                                                      题目: 05455
    class TreeNode:
                                                                     提交人: 2400011037
       def __init__(self, value):
    self.value = value
                                                                      内存: 3648kB
          self.left = None
                                                                      时间: 21ms
          self.right = None
                                                                      语言: Python3
                                                                   提交时间: 2025-04-22 21:51:53
    from collections import deque
    def insert(root, value):
       if root == None:
          return TreeNode (value)
```

if root.value == value:

## M04078: 实现堆结构

手搓实现,http://cs101.openjudge.cn/practice/04078/

类似的题目是 晴问 9.7: 向下调整构建大顶堆, https://sunnywhy.com/sfbj/9/7

思路: 代码:

代码运行截图 (至少包含有"Accepted")

T22161: 哈夫曼编码树

greedy, http://cs101.openjudge.cn/practice/22161/

思路: 代码:

代码运行截图 (至少包含有"Accepted")

## 2. 学习总结和收获

如果发现作业题目相对简单,有否寻找额外的练习题目,如"数算 2025spring 每日选做"、LeetCode、Codeforces、洛谷等网站上的题目。

最近感觉有点不在状态。讲义看过依然有点半知半解,建树目前还是太困难了。还有一门考试 4.30 才结束,这之后有五一假期,我将投入大量的时间恶补。