

Assignment #9: Huffman, BST & Heap

Updated 1834 GMT+8 Apr 15, 2025

2025 spring, Compiled by 王梓航、物理学院

说明：

1. 解题与记录：

对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge，Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora <https://typoraio.cn> 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。

2. **提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。

3. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

LC222.完全二叉树的节点个数

bfs, dfs, binary + greedy, <https://leetcode.cn/problems/count-complete-tree-nodes/>

如果用bfs写是简单级别，其他方法是中级难度。

思路：自己做的时候没有想到，看到题解了才理解了这道题的精髓：充分利用完全二叉树的性质，每次都实现二分，更好地处理问题

代码：


```


class Solution:
    def countNodes(self, root: Optional[TreeNode]) -> int:
        if not root:
            return 0
        def h(root):
            if not root:
                return 0
            return h(root.left)+1
        l,r= h(root.left),h(root.right)
        if l==r:
            return (1<<l) +self.countNodes(root.right)
        else:
            return (1<<r)+self.countNodes(root.left)

```

代码运行截图 (至少包含有"Accepted")

通过 18 / 18 个通过的测试用例

 Hungry NorthcuttqqE 提交于 2025.04.17 15:15

 官方题解

 写题解

LC103.二叉树的锯齿形层序遍历

bfs, <https://leetcode.cn/problems/binary-tree-zigzag-level-order-traversal/>

思路：正常收集，再改成锯齿形即可

代码：

```


class solution:
    from collections import deque
    def zigzagLevelOrder(self, root: Optional[TreeNode]) -> List[List[int]]:
        if not root:
            return []
        b = [[]]
        def f(root,n):
            if len(b)<n+1:
                b.append([])
            b[n].append(root.val)
            if root.left:
                f(root.left,n+1)
            if root.right:
                f(root.right,n+1)
        f(root,0)
        for i in range(len(b)):
            if i%2==1:
                b[i]=b[i][::-1]


```

```
return b
```

代码运行截图 (至少包含有"Accepted")

通过 33 / 33 个通过的测试用例

 Hungry NorthcuttqqE 提交于 2025.04.17 15:45

 官方题解

 写题解

M04080:Huffman编码树

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

思路:

代码:

```
n = int(input())
a = list(map(int, input().split()))
import heapq
heapq.heapify(a)
def f(a,n):
    if n==2:
        return a[0]+a[1]
    x,y=heapq.heappop(a),heapq.heappop(a)
    res = x+y
    heapq.heappush(a,res)
    res +=f(a,n-1)
    return res
print(f(a,n))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n = int(input())
a = list(map(int, input().split()))
import heapq
heapq.heapify(a)
def f(a,n):
    if n==2:
        return a[0]+a[1]
    x,y=heapq.heappop(a),heapq.heappop(a)
    res = x+y
    heapq.heappush(a,res)
    res +=f(a,n-1)
    return res
print(f(a,n))
```

基本信息

#: 48920995
题目: 04080
提交人: 24n2400011481
内存: 3624kB
时间: 21ms
语言: Python3
提交时间: 2025-04-15 22:22:20

M05455: 二叉搜索树的层次遍历

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

思路：按照标准做法处理即可

代码：

```
a = list(map(int,input().split()))
class TreeNode:
    def __init__(self,val=0,left = None,right = None):
        self.val = val
        self.left = left
        self.right = right
def insert(root,n):
    if not root:
        return TreeNode(n)
    if n<root.val:
        root.left = insert(root.left,n)
    else:
        root.right = insert(root.right,n)
    return root
b = set()
node = None
for index in a:
    if index not in b:
        node = insert(node,index)
        b.add(index)
from collections import deque
c = []
d = deque([node])
while d:
    cur = d.popleft()
    c.append(cur.val)
    if cur.left:
        d.append(cur.left)
    if cur.right:
        d.append(cur.right)
print(*c)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
a = list(map(int,input().split()))
class TreeNode:
    def __init__(self,val=0,left = None,right = None):
        self.val = val
        self.left = left
        self.right = right
def insert(root,n):
    if not root:
        return TreeNode(n)
    if n<root.val:
        root.left = insert(root.left,n)
    else:
        root.right = insert(root.right,n)
    return root
b = set()
```

基本信息

#: 48941815
题目: 05455
提交人: 24n2400011481
内存: 3636kB
时间: 19ms
语言: Python3
提交时间: 2025-04-17 22:01:49

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: 按照标准方式构建即可

代码:

```
heap = []
def insert(heap,n):
    heap.append(n)
    i = len(heap)-1
    while i>0 and heap[i]<heap[(i-1)//2]:
        heap[i],heap[(i-1)//2]=heap[(i-1)//2],heap[i]
        i = (i-1)//2
def delete(heap):
    if not heap:
        return None
    elif len(heap)==1:
        print(heap.pop())
        return
    temp = heap.pop()
    ans = heap[0]
    heap[0]=temp
    i = 0
    while True:
        left,right = 2*i+1,2*i+2
        num = i
        if left < len(heap) and heap[left]<heap[num]:
            num = left
        if right<len(heap) and heap[right]<heap[num]:
            num = right
        if num ==i:
            break
```

```

        heap[num], heap[i] = heap[i], heap[num]
        i = num
    print(ans)
n = int(input())
for _ in range(n):
    s = input()
    if s.isdigit():
        delete(heap)
    else:
        s = s.split()
        insert(heap, int(s[1]))

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

heap = []
def insert(heap, n):
    heap.append(n)
    i = len(heap) - 1
    while i > 0 and heap[i] < heap[(i - 1) // 2]:
        heap[i], heap[(i - 1) // 2] = heap[(i - 1) // 2], heap[i]
        i = (i - 1) // 2
def delete(heap):
    if not heap:
        return None
    elif len(heap) == 1:
        print(heap.pop())
        return
    temp = heap.pop()
    ans = heap[0]
    heap[0] = temp
    i = 0

```

基本信息

#: 48942040
 题目: 04078
 提交人: 24n2400011481
 内存: 4656kB
 时间: 521ms
 语言: Python3
 提交时间: 2025-04-17 22:31:03

T22161: 哈夫曼编码树

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

思路: 正常按照要求处理即可

代码:

```

n = int(input())
a = []
b = {}
for _ in range(n):
    s, index = input().split()
    index = int(index)
    b[s] = 1
    a.append((index, s))
import heapq
heapq.heapify(a)

```

```

for _ in range(n-1):
    (i,x) = heapq.heappop(a)
    (j,y) = heapq.heappop(a)
    for index in x:
        b[index]='0'+b[index]
    for index in y:
        b[index] = '1'+b[index]
    heapq.heappush(a,(i+j,x+y))
c = {}
for i,j in b.items():
    c[j]=i
d = set(c.keys())
while True:
    try:
        a = input()
        if a.isdigit():
            ans = ''
            while a:
                for j in range(len(a)):
                    if a[:j+1] in d:
                        ans +=c[a[:j+1]]
                        a = a[j+1:]
                print(ans)
            else:
                ans = ''
                for index in a:
                    ans+=b[index]
                print(ans)
        except EOFError:
            break

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

n = int(input())
a = []
b = {}
for _ in range(n):
    s,index = input().split()
    index = int(index)
    b[s]=''
    a.append((index,s))
import heapq
heapq.heapify(a)

```

基本信息

: 48930748

题目: 22161

提交人: 24n2400011481

内存: 3640kB

时间: 21ms

语言: Python3

提交时间: 2025-04-16 19:36:36

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。

最近在补上每日选做，基本上有些感觉了，但是感觉内容还是很多，要抓紧了