



课时27 二叉树的基本操作

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| B | C | C | A | C | C | C | B | C | A |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| A | B | D | C | B | A | D | B | D | D |
| 21 | 22 | 23 | | | | | | | |
| D | D | D | | | | | | | |

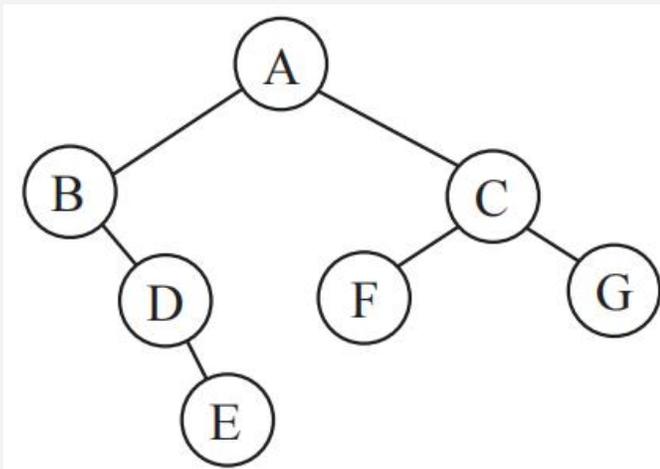
2. 将如图所示的二叉树用列表list方式存储, 下列表示正确的是 (C)

A. ["A", ["B", ["D", "E"]], ["C", "F", "G"]]

B. ["A", ["B", None, ["D", None, "E"]], ["C", "F", "G"]]

C. ["A", ["B", None, ["D", None, ["E", None, None]]], ["C", ["F", None, None], ["G", None, None]]]

D. ["A", ["B", [None, None, None], ["D", None, ["E", None, None]]], ["C", ["F", None, None], ["G", None, None]]]



用列表list表示的树中, 每个节点都用一个单独的列表[d, l, r]表示, 空节点用None表示。其中叶子节点表示为[d, None, None], 注意节点之间的嵌套关系。

5. [2025县域联盟] 某二叉树中有4个节点，其前序遍历结果与后序遍历结果互逆，如abcd与dcba，则该二叉树的中序遍历结果有 (C)

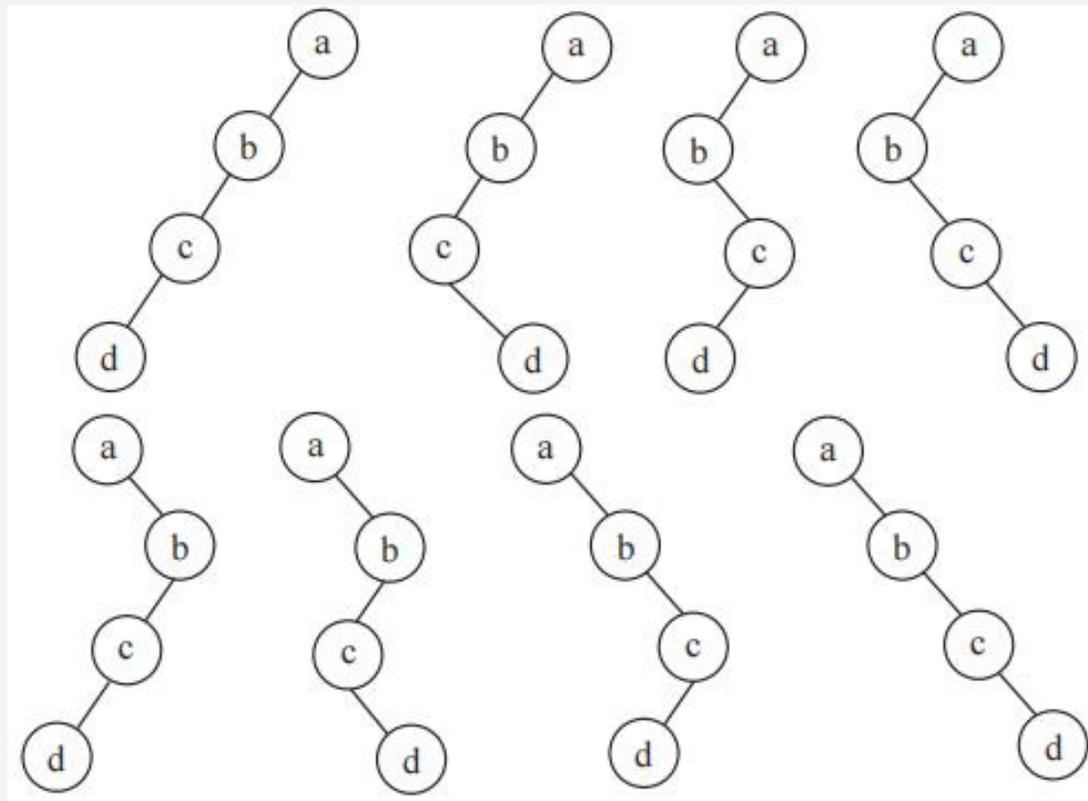
A. 2种

B. 4种

C. 8种

D. 16种

已知前序遍历结果与后序遍历结果互逆，则说明二叉树中所有节点的度均为1，可能的结果如图所示，该二叉树的中序遍历结果有8种。



三、抽象数据类型的描述

6. 定义一个简单的ADT，具体描述如下：

ADT Bag: #创建一个背包类

Bag(self) #创建空背包

is_empty(self) #判断背包是否为空

add_weight(self,wei) #加入重量wei

take_weight(self,wei) #取走重量wei

all_weight(self) #输出背包的总重量

为该抽象数据类型创建一个背包实例bag1并执行如下语句：

bag1.add_weight(8)

bag1.add_weight(2)

bag1.all_weight()

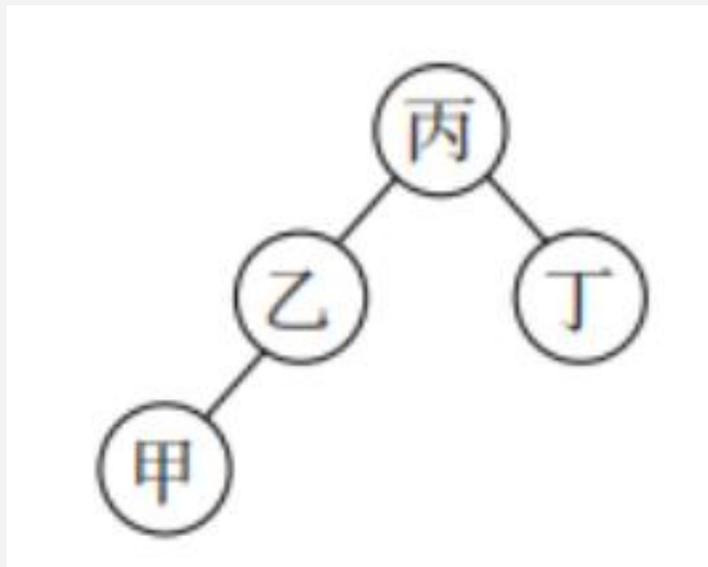
执行该程序段后，输出的结果是 (C)

- A. 8 B. 0 C. 10 D. 6
- 2
- 10

根据代码可知，只有语句 `bag1.all_weight()` 有输出功能，输出的内容为背包 `bag1` 的总重量，在输出之前已向背包 `bag1` 中添加了重量8和2，故输出的总重量为10。

10. 某完全二叉树的中序遍历结果为“甲乙丙丁”，则后序遍历结果是
(A)

- A. 甲乙丁丙
- B. 丙乙甲丁
- C. 甲丁丙乙
- D. 乙丁丙甲



首先画出节点数为4的完全二叉树，然后根据中序遍历结果画出如图所示的完全二叉树，故其后序遍历结果是“甲乙丁丙”。

11. [2025镇海中学检测] 某完全二叉树包含8个节点，其根节点在后序遍历序列、中序遍历序列中的位置序号分别记为 x 、 y ，则 $x-y$ 的值为 (**A**)

A. 3

B. 4

C. 5

D. 6

完全二叉树包含8个节点，则根节点的左子树包含4个节点，右子树包含3个节点。在后序遍历中，根节点是最后一个遍历到的，则位置序号 $x=8$ 。在中序遍历中，先遍历左子树，再遍历根节点，最后遍历右子树，则位置序号 $y=5$ ，故 $x-y$ 的值为3。

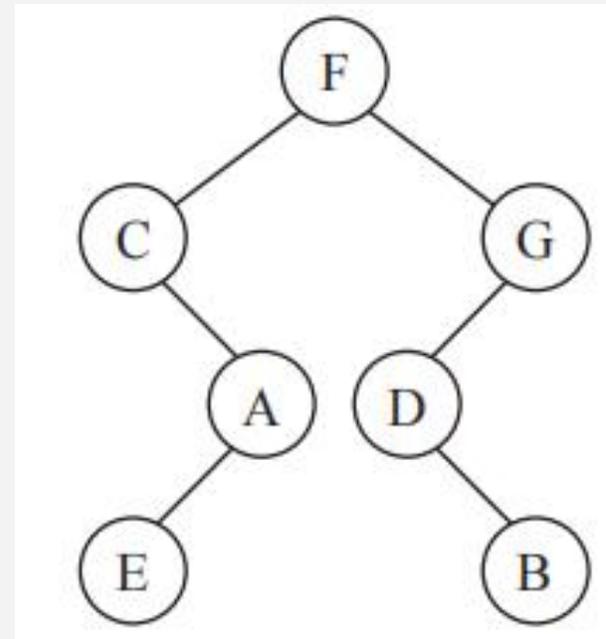
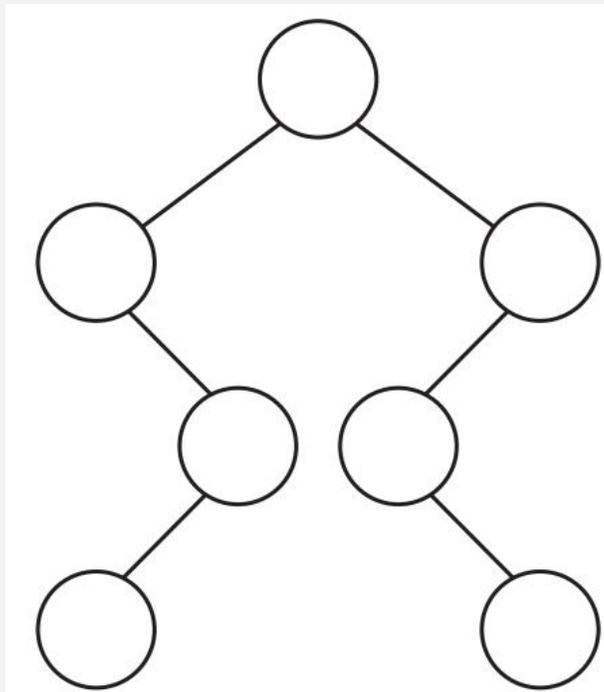
12. [2025浙江联考] 某二叉树的树形结构如图所示, 其后序遍历结果为EACBDGF, 则树中与节点A同层的节点是 (**B**)

A. C

B. D

C. F

D. G



根据二叉树的树形结构和后序遍历结果, 可画出该二叉树如图所示, 与节点A同层的节点是D。

13. [2025湖丽衢联考] 某二叉树的后序遍历结果为ABCDE, 其中节点A和节点B分别是节点C的左孩子和右孩子, 则该二叉树的前序遍历结果可能是

(**D**)

A. CABED

B. DCABE

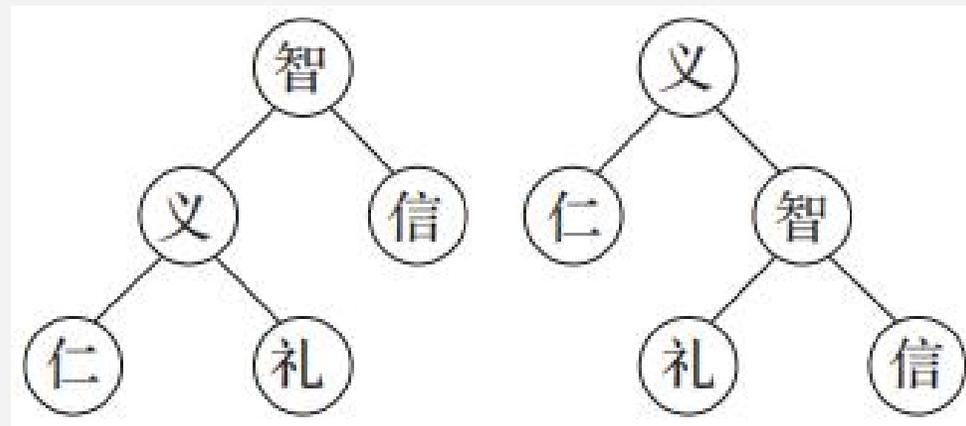
C. EACBD

D. EDCAB

根据后序遍历结果可知, 节点E是根节点, 则前序遍历结果中E一定在第1个位置。节点A和节点B分别是节点C的左孩子和右孩子, 则这3个节点的前序遍历结果一定是CAB。

14. [2025杭州模拟] 某二叉树共有5个节点，其中有3个叶子节点。若中序遍历结果为“仁义礼智信”，则下列说法正确的是 (C)

- A. “仁”的父节点一定是“信”
- B. 根节点一定是“智”
- C. 叶子节点一定是“仁”“礼”“信”
- D. “义”的孩子节点一定是“礼”



根据中序遍历结果，可画出该二叉树的2种可能形态，如图所示。
 “仁”的父节点一定是“义”。根节点可能是“智”或“义”。
 “义”的孩子节点是“仁”“礼”或“仁”“智”。

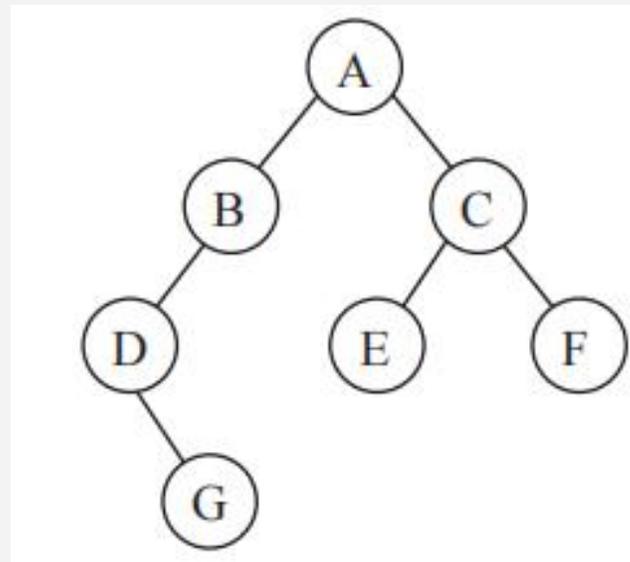
15. [2025金丽衢联考] 某二叉树的中序遍历结果是DGBAECF, 后序遍历结果是GDBEFCA, 下列关于该二叉树的说法, 不正确的是 (B)

A. 该二叉树的前序遍历结果是ABDGCEF

B. 该二叉树是满二叉树

C. 该二叉树有3个叶子节点

D. 节点D是节点G的父节点



根据该二叉树的中序遍历结果和后序遍历结果, 可画出如图所示的二叉树, 故该二叉树不是满二叉树。

16. [2025浙江Z20联盟] 已知某二叉树的前序遍历结果为ABCD, 下列说法正确的是 (A)

- ✓ A. 该二叉树的中序遍历结果可能是BDCA
- B. 度为2的节点数可能与叶子节点数相同
- C. 若该树的深度为4, 则形态只有3种
- D. 若该树为完全二叉树, 则D是分支节点

在任意一棵二叉树中, 若度为2的节点数量为 n_2 , 叶子节点数为 n_0 , 则有 $n_0 = n_2 + 1$ 。若该树的深度为4, 则每一层只有1个节点, 可以是其父节点的左孩子或右孩子, 故形态共有 $2 \times 2 \times 2 = 8$ 种。若该树为完全二叉树, 则D是叶子节点。

17. [2025嵊州统测] 某二叉树的中序遍历结果为ABCDE, 其任意非叶子节点的度为2, 则下列关于该二叉树的说法, 正确的是 (**D**)

A. 该二叉树有4层

B. 该二叉树可能有3种形态

C. 节点A不可能是叶子节点

D. 该二叉树有3个叶子节点

该二叉树共有5个节点, 其任意非叶子节点的度为2, 则该二叉树的结构是: 第1层有1个节点; 第2层有2个节点, 其中1个节点有2个孩子节点, 故该二叉树有3层。该二叉树可能有2种形态。节点A可能是叶子节点。

18. [2025嘉兴模拟] 某完全二叉树总节点数为10, 用数字1~10逐层依次标记各个节点。下列关于该树的说法, 不正确的是 (B)

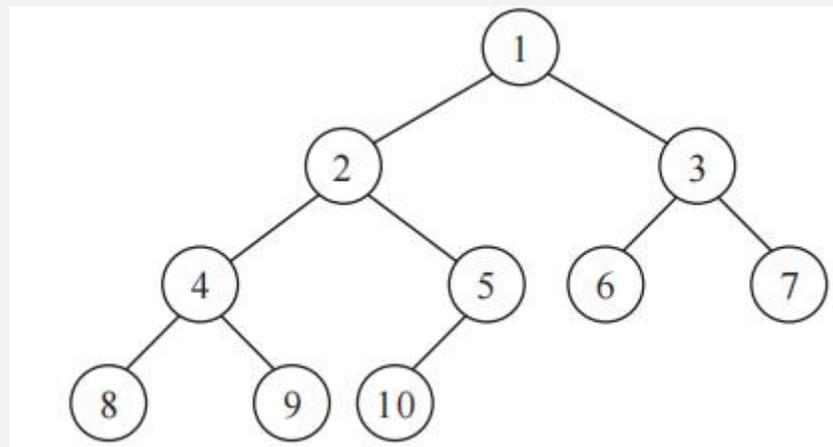
A. 叶子节点数比度为2的节点数多1个

B. 中序遍历的顺序为8-9-4-10-5-2-1-6-3-7

C. 若某节点无左孩子节点, 则此节点也无右孩子节点

D. 用一维数组存储该树时, 所有节点对应的数组元素一定是连续的

该完全二叉树如图所示, 其中序遍历的顺序为8-4-9-2-10-5-1-6-3-7。

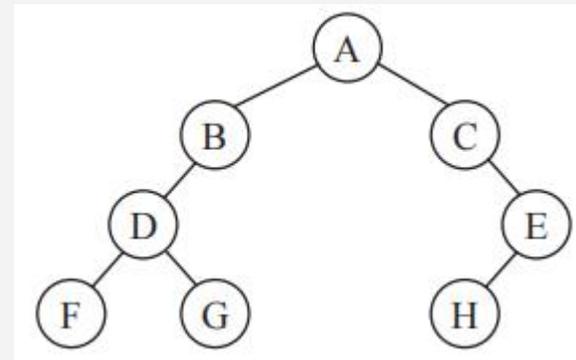


20. 某二叉树用一维数组存储，结构如图所示。

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| A | B | C | D | | | E | F | G | | | | | H | |

下列关于该二叉树的说法，正确的是 (D)

- A. 该二叉树是完全二叉树
- B. 该二叉树中度为2的节点有3个
- C. 该二叉树的前序遍历结果是ABDFGCHE
- ✓ D. 节点C是节点E的父节点

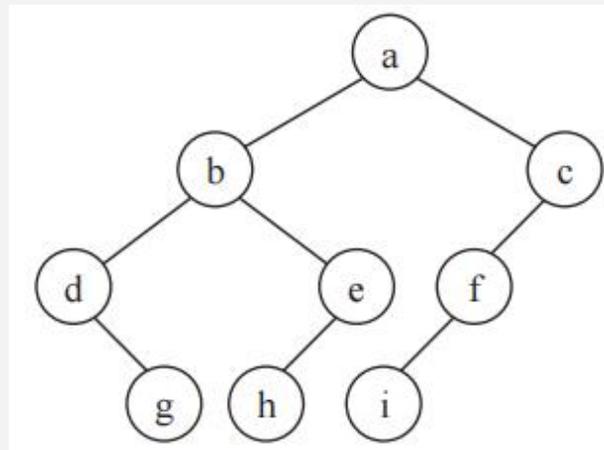


根据一维数组画出如图所示的二叉树。该二叉树不是完全二叉树。该二叉树中度为2的节点有2个。该二叉树的前序遍历结果为ABDFGCHEH。

21. [2025余姚中学检测] 用如图所示的数组表示一棵二叉树。

下列说法正确的是 (D)

- A. 这是一棵完全二叉树
- B. 这棵树的叶子节点比度为1的节点多1个
- C. 节点g和h是兄弟节点
- ✓ D. 使用前序遍历和中序遍历方式查找节点e, 访问节点个数相同



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| a | b | c | d | e | f | | | g | h | | i |

根据数组表示的二叉树, 可画出二叉树的结构如图所示。该二叉树不是完全二叉树。该树的叶子节点个数为3, 度为1的节点个数为4。节点g和h不是兄弟节点。使用前序遍历和中序遍历方式查找节点e, 访问节点个数都为5。

24. 排序二叉树是指符合如下3个条件的二叉树：

- ①若左子树不空，则左子树上所有节点的值均小于它的根节点的值。
- ②若右子树不空，则右子树上所有节点的值均大于它的根节点的值。
- ③左、右子树也分别为二叉排序树。

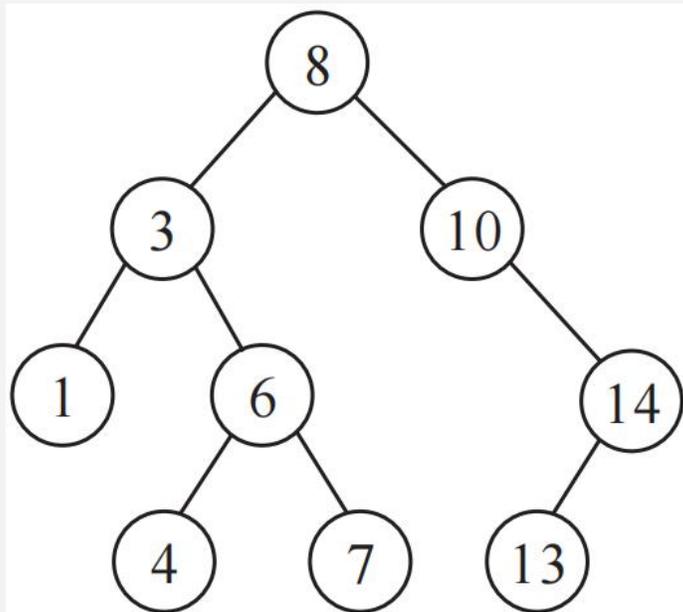


图 a

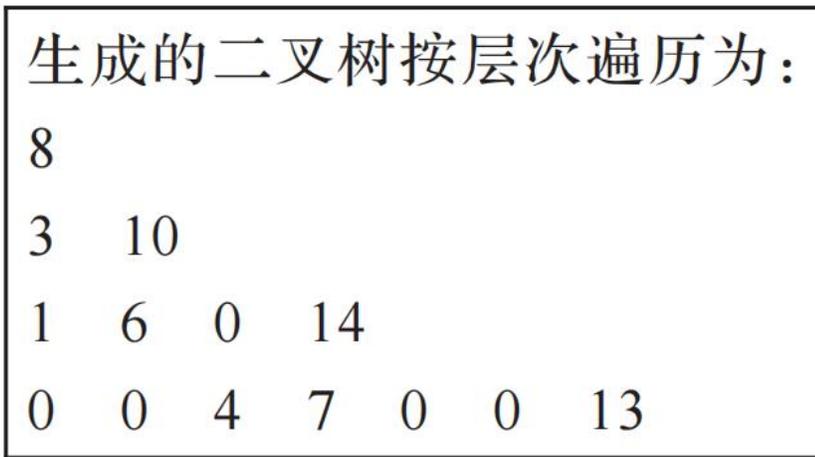


图 b

(1) 定义如下函数`create_tree()`，实现使用列表`data`提供的非零正整数（列表元素个数小于等于10）构建排序二叉树`tree`。例如`data=[8, 10, 3, 6, 7, 4, 14, 13, 1]`，构成的排序二叉树如图a所示，该二叉树的一维数组存储形式`tree=[8, 3, 10, 1, 6, "", 14, "", "", 4, 7, 13]`。请在划线处填入合适的代码。

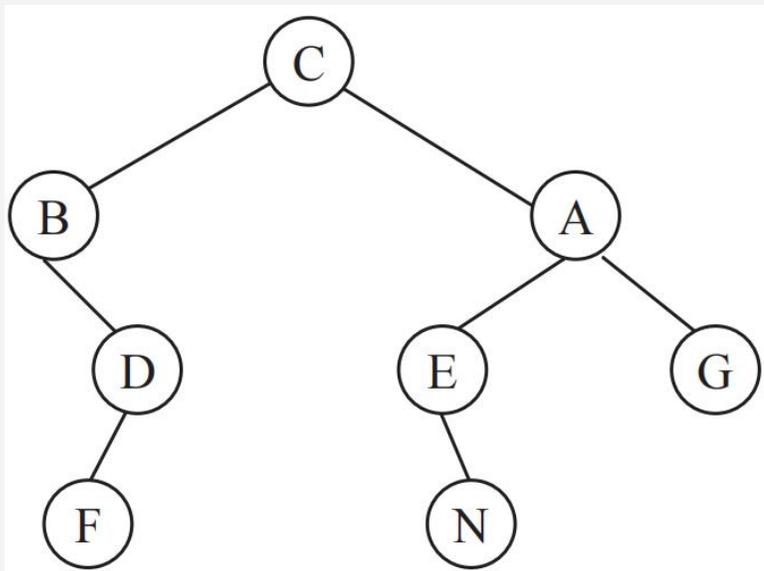
```
def create_tree(tree, data):  
    for i in range(len(data)):  
        depth=0  
        if i==0:  
            _____ ① tree[0]=data[0]  
        else:  
            while tree[depth]!=0:  
                if data[i]>tree[depth]:  
                    _____ ② depth=depth*2+2  
                else:  
                    depth=depth*2+1  
            tree[depth]=data[i]
```

(2) 编写如下Python程序, 实现以层次遍历的方法遍历该排序二叉树, 其中无节点的位置用0表示。程序运行结果如图b所示, 请在划线处填入合适的代码。

```
btree=[0]*20
'''生成data数据序列, 例如[8, 10, 3, 6, 7, 4, 14, 13, 1], 代码略'''
create_tree(btree, data)
print("生成的二叉树按层次遍历为:")
last=len(btree)-1
while last>=0 and btree[last]==0:
    last-=1
i=0;k=1
while _____① i<=last:
    for j in range(i, i+k):
        if j>last:
            break
        else:
            print(btree[j], end=" ")
    print() #换行
    i+=k
    _____② k=k*2
```

(3) 用数据序列[5, 8, 4, 3, 7, 6]生成排序二叉树后, 其中序遍历结果是 3,4,5,6,7,8。

26. 有二叉树如图所示。



下标
数据

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| C | B | A | | D | E | G | | | F | | | N |

- 请写出该二叉树的数组表示。
- 请写出该二叉树的链表表示。

| | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|----|----|----|
| 下标 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 数据 | | | | | | | | | | | | | |

