

4. In a complete binary tree with 2024 nodes, how many leaf nodes are there?

- A. 1011
- B. 1012
- C. 1013
- D. 1014

$2^{10} = 1024 \leq 2024 < 2^{11} = 2048$ , 共有11层,  $2024 - 1023 = 1001$ , 距离满11层1024少了23个结点. 所有第10层有  $\lfloor 23 \rfloor = 11$ , 总共有  $1001 + 11 = 1012$  个结点.

10. When performing a delete operation in a binary search tree (BST), if the node to be deleted has two children, what is the **most common operation**?

- A. Replace it with the smallest node in its right subtree and recursively delete that node.
- B. Replace it with the smallest node in its left subtree and recursively delete that node.
- C. Replace it with the largest node in its right subtree and recursively delete that node.
- D. Replace it with the largest node in its left subtree and recursively delete that node.

- **BST性质**要求右子树的节点值大于当前节点值，因此删除节点时，常用右子树中最小的节点（后继节点）替换，保持 tree 的有序性。所以，递归删除后继节点是实现删除的标准步骤，因后继节点的值已经被替换，需要从原位置移除它。
- 课件10中有对于三种删除情况的具体讲解，分别对应(1)没有子结点，(2)有1个子结点，本题是情况(3)有2个子结点。

事实上，B会破坏BST性质，C不常用因为右子树的最大节点可能远离被删除节点，D实际上是用前驱节点替换的方式，但并非最常用。

8. In a binary tree, if node  $m$  is an ancestor of node  $n$ , which traversal can be used to find the path from  $m$  to  $n$ ?

- A. Level-order traversal
- B. Preorder traversal
- C. Inorder traversal
- D. Postorder traversal

这道题几乎是所有同学都做错了的一道题。

许多选择前序遍历的同学可能最初会直接想到：前序遍历的特点是“先根后子”，可能直觉上，既然根节点  $m$  会在最开始访问，接下来依次访问其子节点，前序遍历似乎可以很容易找到从  $m$  到  $n$  的路径？

但事实上，前序遍历访问完一个节点后，若进入其他子树（尤其是右子树），路径信息可能会丢失。

后序遍历是从底向上的遍历方式，当你访问到目标节点  $n$  时，可以通过递归函数的回溯或非递归的栈信息，逐步找到路径上的祖先节点，栈中的节点正好记录了路径上的所有父节点；而对于非递归，栈保存了从根节点到目标节点的父节点路径，当你访问到目标节点  $n$ ，栈中又恰好记录从根  $m$  到  $n$  的路径节点。

所以这里并不只是考核了你对四种遍历方法顺序的基本了解，实际上你应该知道后序遍历的退回特性和非递归方法中的栈记录。

至于A，缺乏层级之间的父子关系，无法构建路径；C则是差得最远的，因为中序甚至访问顺序与路径不一致，无法记录路径结构。

## AVL和解析树的构造

4. Insert the nodes [34, 23, 15, 98, 115, 28, 107] sequentially into an initially empty balanced binary search tree (AVL Tree). Draw the resulting tree after each insertion. (No code needed, only diagrams)

同样，根据题目要求需要具有一定步骤：