Second Smallest In Tree

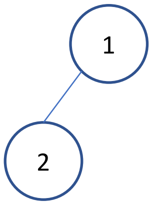
Find the second smallest value in a binary tree, if one exists.

```
class TreeNode
{
     public:
          TreeNode *left, *right;
          int value;
}; // end class TreeNode
```

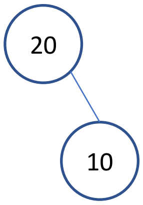
Write a function
`bool secondsmallest(TreeNode *root, int &val)`
where
`root` is a pointer to the root of the tree (possibly NULL)
and returns `true` if and only if there exists at least two nodes in the tree in which `val` is set to the second smallest value in the tree,
otherwise return `false` and `val` is ignored.

File you must submit: `soln_func.cc`

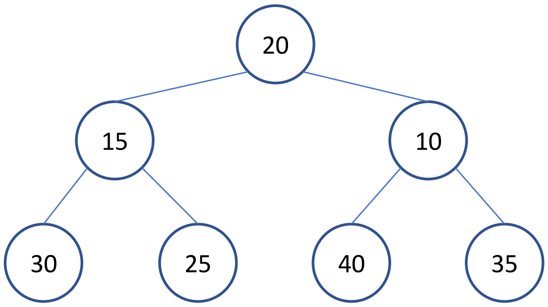
Examples:



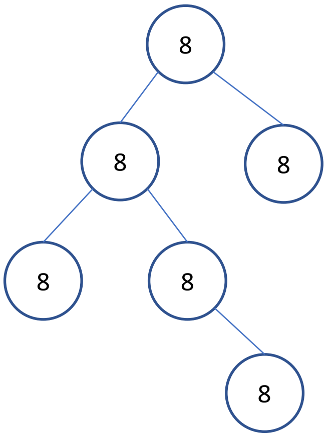
Returns: `true`, sets val=1



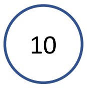
Returns: `true`, sets val=10



Returns: `true`, sets val=15



Returns: `true`, sets val=8



Returns: `false`, val ignored
Explanation: Tree has only one node.