**Problem #3 - Debug Calendar Design**

from typing import Optional  
class Node:  
 def \_\_init\_\_(self, start: int, end: int):  
 self.start: int = start  
 self.end: int = end  
 self.left\_child: Optional[Node] = None  
 self.right\_child: Optional[Node] = None  
 def insert(self, node: 'Node') -> bool:  
 # Check for overlap  
 if node.start < self.end and node.end > self.start:  
 return False # Overlap detected  
  
 # Insert in the right subtree  
 if node.start >= self.end:  
 if not self.right\_child:  
 self.right\_child = node  
 return True  
 return self.right\_child.insert(node)  
  
 # Insert in the left subtree  
 if node.end <= self.start:  
 if not self.left\_child:  
 self.left\_child = node  
 return True  
 return self.left\_child.insert(node)  
 return False # This case should never occur logically  
class Calendar:  
 def \_\_init\_\_(self):  
 self.root: Optional[Node] = None  
 def book(self, start: int, end: int) -> bool:  
 new\_node = Node(start=start, end=end)  
 if self.root is None:  
 self.root = new\_node  
 return True  
 return self.root.insert(new\_node)  
calendar = Calendar()  
print(calendar.book(5, 10))

print(calendar.book(8, 13))

print(calendar.book(10, 15))

**output:**

True

False

True