두 노드 간 값의 차이가 가장 작은 노드의 값의 차이를 출력하라.

Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6

1.리스트로 변환

```
import sys

class Solution:
    def minDiffInBST(self, root: TreeNode) -> int:
        def preOrder(node):
            if node == None:
                return
            preOrder(node.left)
            self.arr.append(node.val)
            preOrder(node.right)

        result = sys.maxsize
        self.arr = []
        preOrder(root)
        self.arr.sort()
        for i in range(len(self.arr) - 1):
            dif = abs(self.arr[i] - self.arr[i + 1])
            if dif < result:
                result = dif
        return result
```

2.중위 순회

```
import sys

class Solution:
    prev = -sys.maxsize
    result = sys.maxsize

    def minDiffInBST(self, root: TreeNode) -> int:
        if root.left:
            self.minDiffInBST(root.left)
        self.result = min(self.result, root.val - self.prev)
        self.prev = root.val
        if root.right:
            self.minDiffInBST(root.right)
        return self.result
```

- 재귀

```
import sys

class Solution:
    def minDiffInBST(self, root: TreeNode) -> int:
        prev = -sys.maxsize
```

```
result = sys.maxsize
stack = []
node = root
while stack or node:
    while node:
        stack.append(node)
        node = node.left
    node = stack.pop()
    result = min(result, node.val - prev)
    prev = node.val
    node = node.right
return result
```

▸ 반복