# 110526005-林季陽-HW5-Code Coverage and JUnit Test Result

## 以下為本次作業 JUnit 中，Code Coverage 與 JUnit Test的結果圖與說明

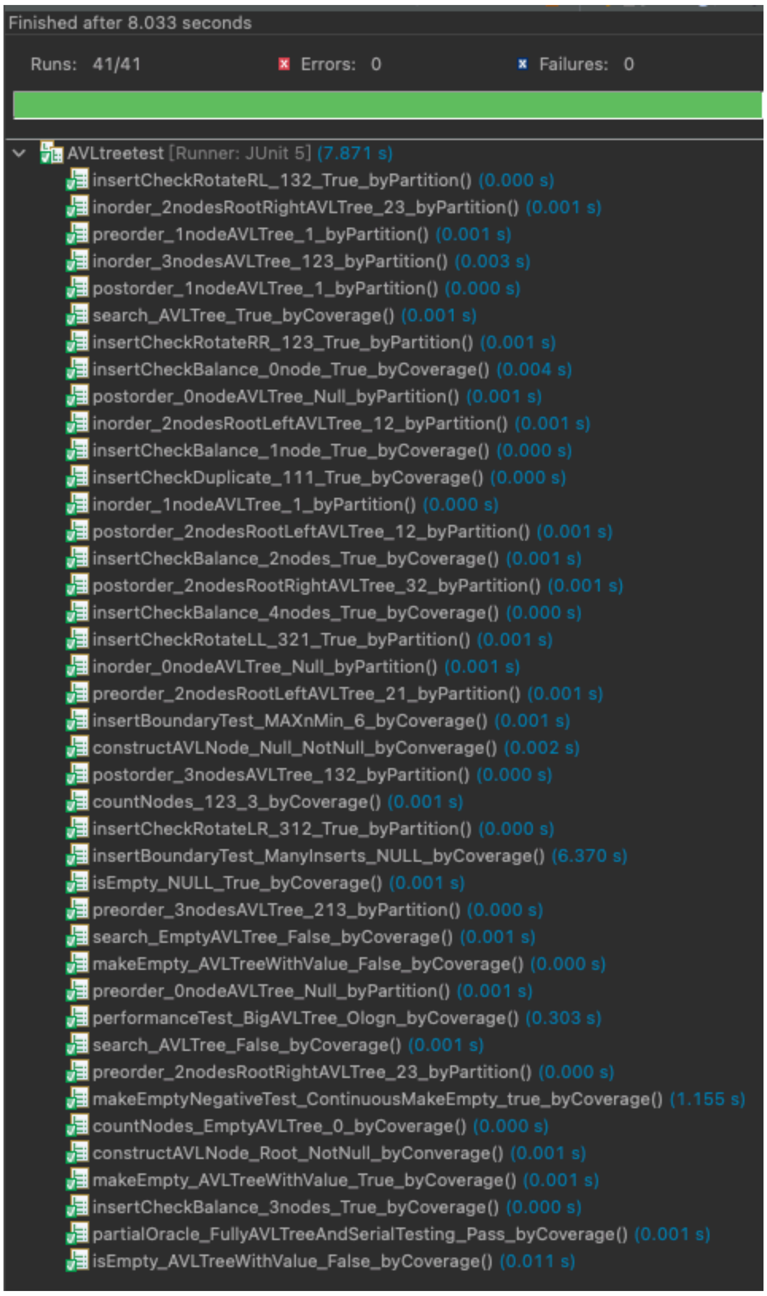
---

1. JUnit Test Result

執行：41/41

錯誤：0

失敗：0



```
Finished after 8.033 seconds

Runs:  41/41          Errors:  0          Failures:  0

AVLtreetest [Runner: JUnit 5] (7.871 s)
    insertCheckRotateRL_132_True_byPartition() (0.000 s)
    inorder_2nodesRootRightAVLTree_23_byPartition() (0.001 s)
    preorder_1nodeAVLTree_1_byPartition() (0.001 s)
    inorder_3nodesAVLTree_123_byPartition() (0.003 s)
    postorder_1nodeAVLTree_1_byPartition() (0.000 s)
    search_AVLTree_True_byCoverage() (0.001 s)
    insertCheckRotateRR_123_True_byPartition() (0.001 s)
    insertCheckBalance_0node_True_byCoverage() (0.004 s)
    postorder_0nodeAVLTree_Null_byPartition() (0.001 s)
    inorder_2nodesRootLeftAVLTree_12_byPartition() (0.001 s)
    insertCheckBalance_1node_True_byCoverage() (0.000 s)
    insertCheckDuplicate_111_True_byCoverage() (0.000 s)
    inorder_1nodeAVLTree_1_byPartition() (0.000 s)
    postorder_2nodesRootLeftAVLTree_12_byPartition() (0.001 s)
    insertCheckBalance_2nodes_True_byCoverage() (0.001 s)
    postorder_2nodesRootRightAVLTree_32_byPartition() (0.001 s)
    insertCheckBalance_4nodes_True_byCoverage() (0.000 s)
    insertCheckRotateLL_321_True_byPartition() (0.001 s)
    inorder_0nodeAVLTree_Null_byPartition() (0.001 s)
    preorder_2nodesRootLeftAVLTree_21_byPartition() (0.001 s)
    insertBoundaryTest_MAXnMin_6_byCoverage() (0.001 s)
    constructAVLNode_Null_NotNull_byConverage() (0.002 s)
    postorder_3nodesAVLTree_132_byPartition() (0.000 s)
    countNodes_123_3_byCoverage() (0.001 s)
    insertCheckRotateLR_312_True_byPartition() (0.000 s)
    insertBoundaryTest_ManyInserts_NULL_byCoverage() (6.370 s)
    isEmpty_NULL_True_byCoverage() (0.001 s)
    preorder_3nodesAVLTree_213_byPartition() (0.000 s)
    search_EmptyAVLTree_False_byCoverage() (0.001 s)
    makeEmpty_AVLTreeWithValue_False_byCoverage() (0.000 s)
    preorder_0nodeAVLTree_Null_byPartition() (0.001 s)
    performanceTest_BigAVLTree_Ologn_byCoverage() (0.303 s)
    search_AVLTree_False_byCoverage() (0.001 s)
    preorder_2nodesRootRightAVLTree_23_byPartition() (0.000 s)
    makeEmptyNegativeTest_ContinuousMakeEmpty_true_byCoverage() (1.155 s)
    countNodes_EmptyAVLTree_0_byCoverage() (0.000 s)
    constructAVLNode_Root_NotNull_byConverage() (0.001 s)
    makeEmpty_AVLTreeWithValue_True_byCoverage() (0.001 s)
    insertCheckBalance_3nodes_True_byCoverage() (0.000 s)
    partialOracle_FullyAVLTreeAndSerialTesting_Pass_byCoverage() (0.001 s)
    isEmpty_AVLTreeWithValue_False_byCoverage() (0.011 s)
```

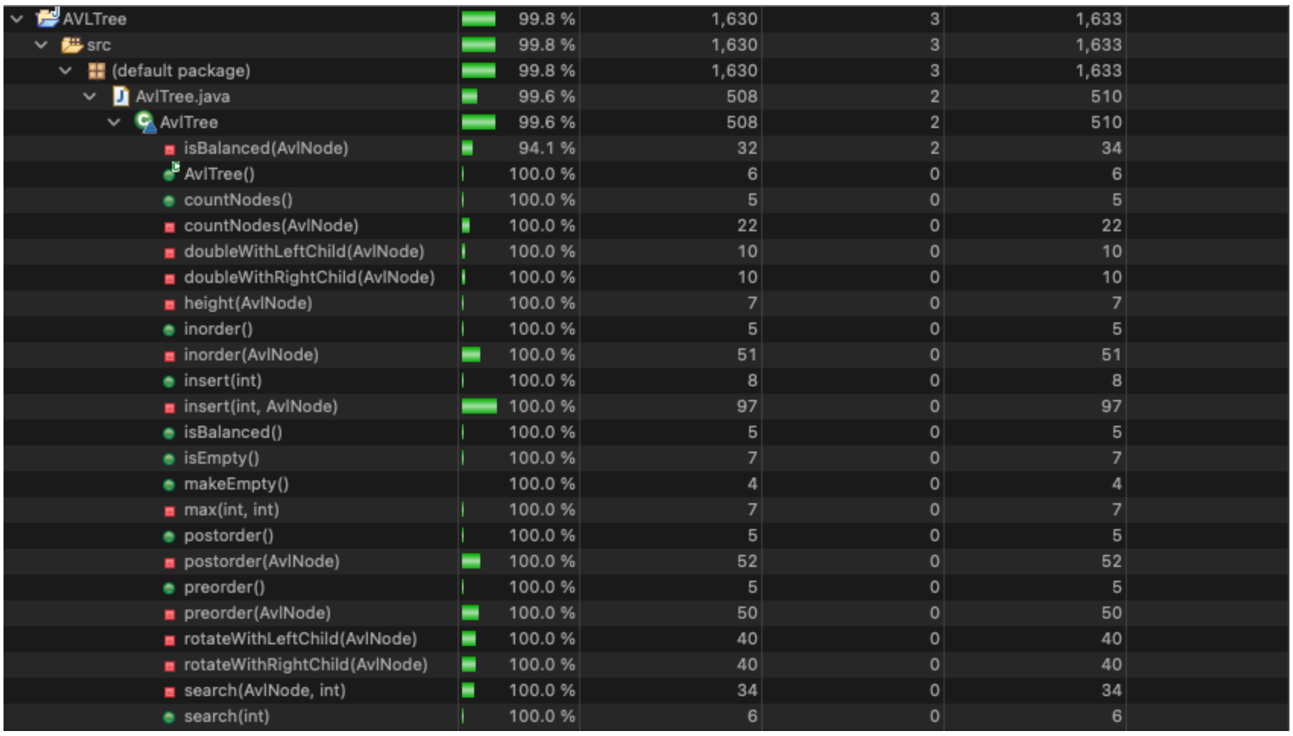
## 2. Code Coverage

Coverage: 99.8% (1630/1633)

本次作業中，於AvlTree.java之中的所有public method都有完整測試到，包含：建構子、insert中的Rotation、Duplicate、比較後放在左子樹右子樹，其他 method 各個分支的情況也都有測試到，像是Null等相關都有進行測試。
而在測試public method的過程中也會連同private method的一起進行測試。
以下為本次Code Coverage的結果與三個未測到部分的說明：

AVLTree.java

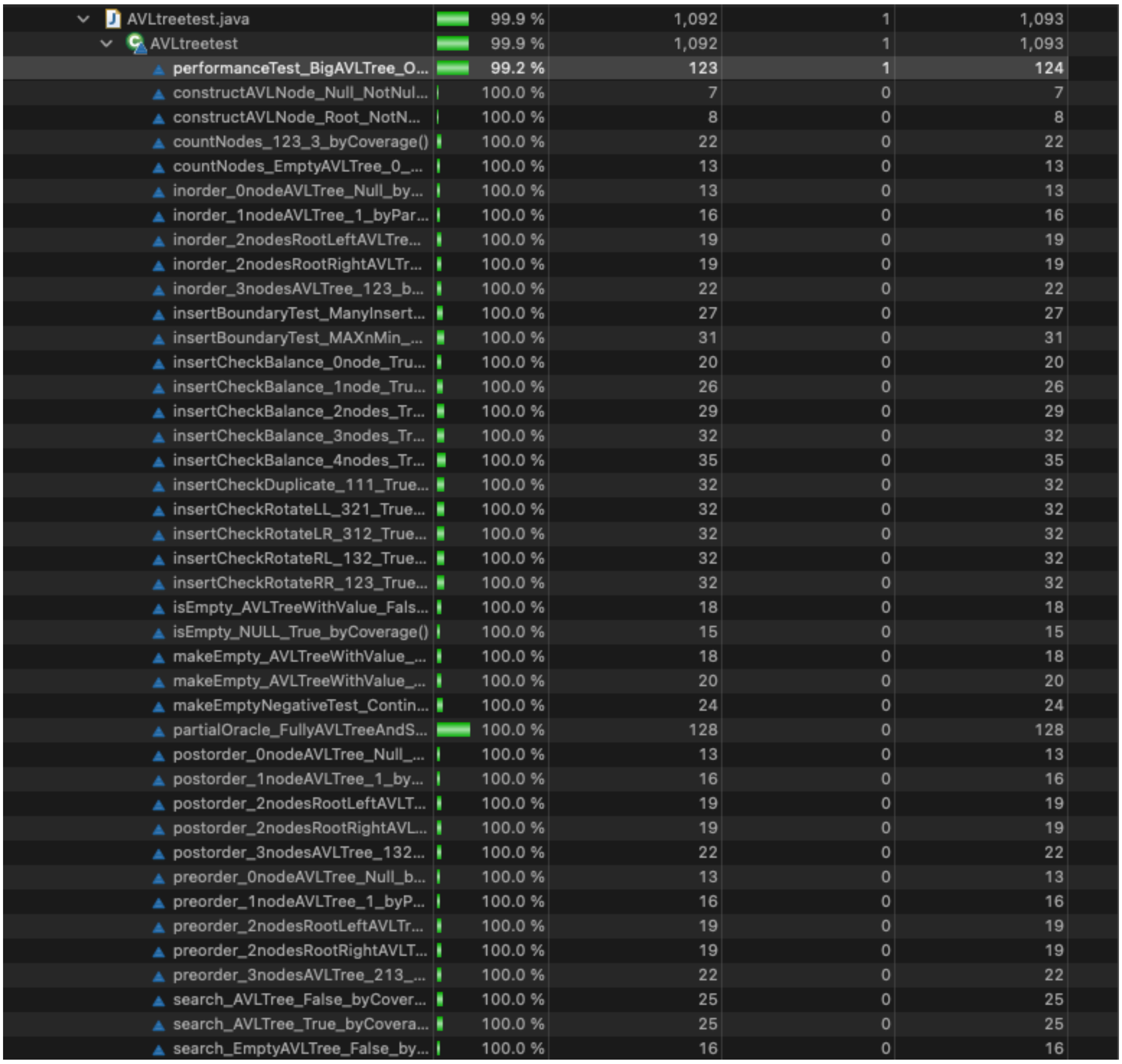
isBalanced 為自行加入來檢查子樹高度是否有平衡的函式，在本次作業中自己希望在不更改原本 class 中任何現有的 public/private method 的精神下，沒辦法走到 子樹 height 差距大於等於2的情況，因為insert就會強制作 rotation，所以這裡有兩個 Missed Instructions為isBalanced之中的。



| | | | | |
|---|---|---|---|---|
| AVLTree | 99.8 % | 1,630 | 3 | 1,633 |
| src | 99.8 % | 1,630 | 3 | 1,633 |
| (default package) | 99.8 % | 1,630 | 3 | 1,633 |
| AvlTree.java | 99.6 % | 508 | 2 | 510 |
| AvlTree | 99.6 % | 508 | 2 | 510 |
| isBalanced(AvlNode) | 94.1 % | 32 | 2 | 34 |
| AvlTree() | 100.0 % | 6 | 0 | 6 |
| countNodes() | 100.0 % | 5 | 0 | 5 |
| countNodes(AvlNode) | 100.0 % | 22 | 0 | 22 |
| doubleWithLeftChild(AvlNode) | 100.0 % | 10 | 0 | 10 |
| doubleWithRightChild(AvlNode) | 100.0 % | 10 | 0 | 10 |
| height(AvlNode) | 100.0 % | 7 | 0 | 7 |
| inorder() | 100.0 % | 5 | 0 | 5 |
| inorder(AvlNode) | 100.0 % | 51 | 0 | 51 |
| insert(int) | 100.0 % | 8 | 0 | 8 |
| insert(int, AvlNode) | 100.0 % | 97 | 0 | 97 |
| isBalanced() | 100.0 % | 5 | 0 | 5 |
| isEmpty() | 100.0 % | 7 | 0 | 7 |
| makeEmpty() | 100.0 % | 4 | 0 | 4 |
| max(int, int) | 100.0 % | 7 | 0 | 7 |
| postorder() | 100.0 % | 5 | 0 | 5 |
| postorder(AvlNode) | 100.0 % | 52 | 0 | 52 |
| preorder() | 100.0 % | 5 | 0 | 5 |
| preorder(AvlNode) | 100.0 % | 50 | 0 | 50 |
| rotateWithLeftChild(AvlNode) | 100.0 % | 40 | 0 | 40 |
| rotateWithRightChild(AvlNode) | 100.0 % | 40 | 0 | 40 |
| search(AvlNode, int) | 100.0 % | 34 | 0 | 34 |
| search(int) | 100.0 % | 6 | 0 | 6 |

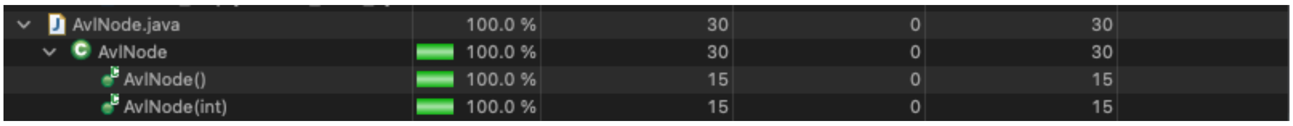
AVLTreeTest.java

另外一個 missed instruction 則在測試檔中的performanceTest_BigAVLTree_Ologn_byCoverage中出現 ，因為最後compareResult時，在O(log n) 的情況下，永遠不會執行到 rate> result 這條branch，所以這裡也有一個沒測到。



| | | | | |
|---|---|---|---|---|
| AVLtreetest.java | 99.9 % | 1,092 | 1 | 1,093 |
| AVLtreetest | 99.9 % | 1,092 | 1 | 1,093 |
| performanceTest_BigAVLTree_O... | 99.2 % | 123 | 1 | 124 |
| constructAVLNode_Null_NotNul... | 100.0 % | 7 | 0 | 7 |
| constructAVLNode_Root_NotN... | 100.0 % | 8 | 0 | 8 |
| countNodes_123_3_byCoverage() | 100.0 % | 22 | 0 | 22 |
| countNodes_EmptyAVLTree_0_... | 100.0 % | 13 | 0 | 13 |
| inorder_0nodeAVLTree_Null_by... | 100.0 % | 13 | 0 | 13 |
| inorder_1nodeAVLTree_1_byPar... | 100.0 % | 16 | 0 | 16 |
| inorder_2nodesRootLeftAVLTre... | 100.0 % | 19 | 0 | 19 |
| inorder_2nodesRootRightAVLTr... | 100.0 % | 19 | 0 | 19 |
| inorder_3nodesAVLTree_123_b... | 100.0 % | 22 | 0 | 22 |
| insertBoundaryTest_ManyInsert... | 100.0 % | 27 | 0 | 27 |
| insertBoundaryTest_MAXnMin_... | 100.0 % | 31 | 0 | 31 |
| insertCheckBalance_0node_Tru... | 100.0 % | 20 | 0 | 20 |
| insertCheckBalance_1node_Tru... | 100.0 % | 26 | 0 | 26 |
| insertCheckBalance_2nodes_Tr... | 100.0 % | 29 | 0 | 29 |
| insertCheckBalance_3nodes_Tr... | 100.0 % | 32 | 0 | 32 |
| insertCheckBalance_4nodes_Tr... | 100.0 % | 35 | 0 | 35 |
| insertCheckDuplicate_111_True... | 100.0 % | 32 | 0 | 32 |
| insertCheckRotateLL_321_True... | 100.0 % | 32 | 0 | 32 |
| insertCheckRotateLR_312_True... | 100.0 % | 32 | 0 | 32 |
| insertCheckRotateRL_132_True... | 100.0 % | 32 | 0 | 32 |
| insertCheckRotateRR_123_True... | 100.0 % | 32 | 0 | 32 |
| isEmpty_AVLTreeWithValue_Fals... | 100.0 % | 18 | 0 | 18 |
| isEmpty_NULL_True_byCoverage() | 100.0 % | 15 | 0 | 15 |
| makeEmpty_AVLTreeWithValue_... | 100.0 % | 18 | 0 | 18 |
| makeEmpty_AVLTreeWithValue_... | 100.0 % | 20 | 0 | 20 |
| makeEmptyNegativeTest_Contin... | 100.0 % | 24 | 0 | 24 |
| partialOracle_FullyAVLTreeAndS... | 100.0 % | 128 | 0 | 128 |
| postorder_0nodeAVLTree_Null_... | 100.0 % | 13 | 0 | 13 |
| postorder_1nodeAVLTree_1_by... | 100.0 % | 16 | 0 | 16 |
| postorder_2nodesRootLeftAVLT... | 100.0 % | 19 | 0 | 19 |
| postorder_2nodesRootRightAVL... | 100.0 % | 19 | 0 | 19 |
| postorder_3nodesAVLTree_132... | 100.0 % | 22 | 0 | 22 |
| preorder_0nodeAVLTree_Null_b... | 100.0 % | 13 | 0 | 13 |
| preorder_1nodeAVLTree_1_byP... | 100.0 % | 16 | 0 | 16 |
| preorder_2nodesRootLeftAVLTr... | 100.0 % | 19 | 0 | 19 |
| preorder_2nodesRootRightAVLT... | 100.0 % | 19 | 0 | 19 |
| preorder_3nodesAVLTree_213_... | 100.0 % | 22 | 0 | 22 |
| search_AVLTree_False_byCover... | 100.0 % | 25 | 0 | 25 |
| search_AVLTree_True_byCovera... | 100.0 % | 25 | 0 | 25 |
| search_EmptyAVLTree_False_by... | 100.0 % | 16 | 0 | 16 |

AvlNode.java
建構子中兩個都有完整執行到。



| | | | | |
|---|---|---|---|---|
| AvlNode.java | 100.0 % | 30 | 0 | 30 |
| AvlNode | 100.0 % | 30 | 0 | 30 |
| AvlNode() | 100.0 % | 15 | 0 | 15 |
| AvlNode(int) | 100.0 % | 15 | 0 | 15 |