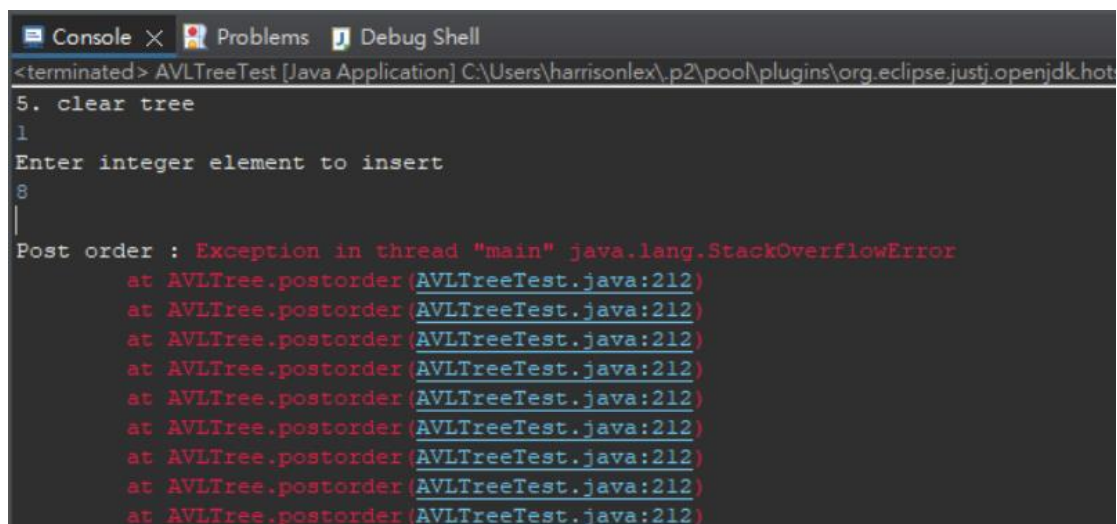


# HW1\_109403021

- Bug1: rotateWithLeftChild()這個 function 內部邏輯錯誤還有使用此 function 時參數輸入錯誤
- Bug2: rotateWithRightChild()這個 function 內部邏輯錯誤還有使用此 function 時參數輸入錯誤

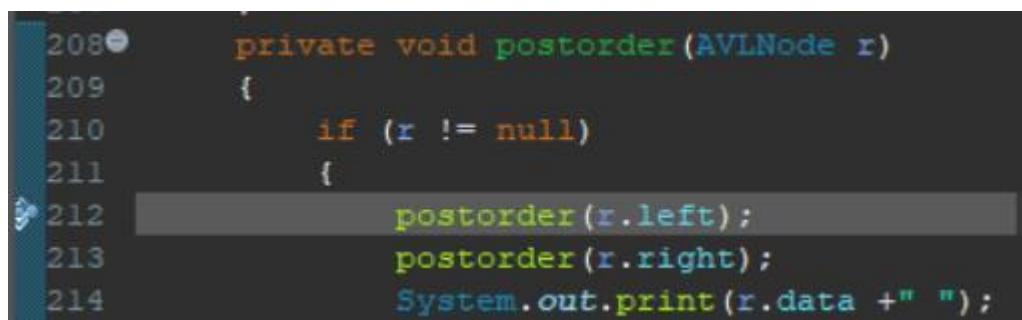
## Debug 流程

我按照表一先執行看看，結果在插入 8 時就遇到錯誤



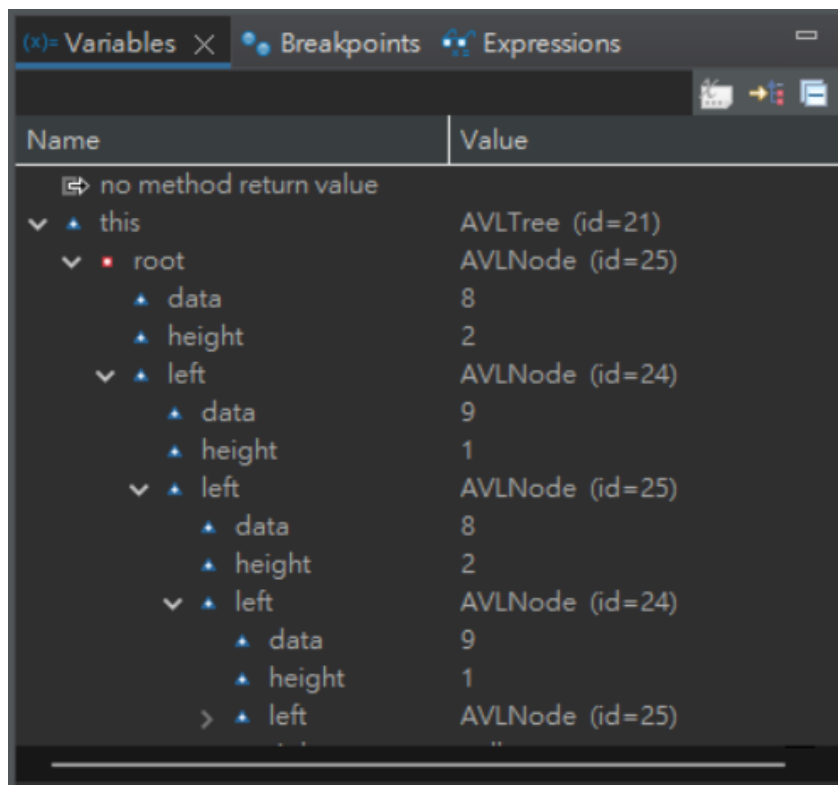
```
Console × Problems Debug Shell
<terminated> AVLTreeTest [Java Application] C:\Users\harrisonlex\p2\pool\plugins\org.eclipse.justj.openjdk.hot
5. clear tree
1
Enter integer element to insert
8
|
Post order : Exception in thread "main" java.lang.StackOverflowError
    at AVLTree.postorder (AVLTreeTest.java:212)
    at AVLTree.postorder (AVLTreeTest.java:212)
    at AVLTree.postorder (AVLTreeTest.java:212)
    at AVLTree.postorder (AVLTreeTest.java:212)
    at AVLTree.postorder (AVLTreeTest.java:212)
    at AVLTree.postorder (AVLTreeTest.java:212)
    at AVLTree.postorder (AVLTreeTest.java:212)
    at AVLTree.postorder (AVLTreeTest.java:212)
    at AVLTree.postorder (AVLTreeTest.java:212)
```

所以我先依照上圖在 212 行設一個 break point



```
208 private void postorder (AVLNode r)
209 {
210     if (r != null)
211     {
212         postorder (r.left);
213         postorder (r.right);
214         System.out.print(r.data + " ");
```

結果發現樹在執行 212 行前已經長得不正確，所以應該是 insert 時就出錯了



Name	Value
no method return value	
▼ this	AVLTree (id=21)
▼ root	AVLNode (id=25)
data	8
height	2
▼ left	AVLNode (id=24)
data	9
height	1
▼ left	AVLNode (id=25)
data	8
height	2
▼ left	AVLNode (id=24)
data	9
height	1
> left	AVLNode (id=25)

所以我在 insert 的 function 中設了一個 breakpoint，然後到裡面一直按 F6 看哪裡出問題

```
67      /* Function to insert data recursively */
68      private AVLNode insert(int x, AVLNode t)
69      {
70          if (t == null)
71              t = new AVLNode(x);
72          else if (x < t.data)
73          {
74              t.left = insert(x, t.left);
75              if (height(t.left) - height(t.right) == 2)
76                  if (x < t.left.data)
77                      t = rotateWithLeftChild(t.left);
78                  else
79                      t = doubleWithLeftChild(t);
80          }
81          else if (x > t.data)
82          {
83              t.right = insert(x, t.right);
84              if (height(t.right) - height(t.left) == 2)
```

結果發現樹是在執行完 77 行 rotate 的 function 變不正確

```
73 {
74     t.left = insert( x, t.left );
75     if( height( t.left ) - height( t.right ) == 2 )
76         if( x < t.left.data )
77             t = rotateWithLeftChild( t.left );
78         else
79             t = doubleWithLeftChild( t );
80     }
81     else if( x > t.data )
82     {
83         t.right = insert( x, t.right );
84         if( height( t.right ) - height( t.left ) == 2 )
85             if( x > t.right.data )
86                 t = rotateWithRightChild( t.right );
87             else
88                 t = doubleWithRightChild( t );
89     }
90     else{
91         return t;
92     }
93
94     t.height = max( height( t.left ), height( t.right ) ) + 1;
95 }
```

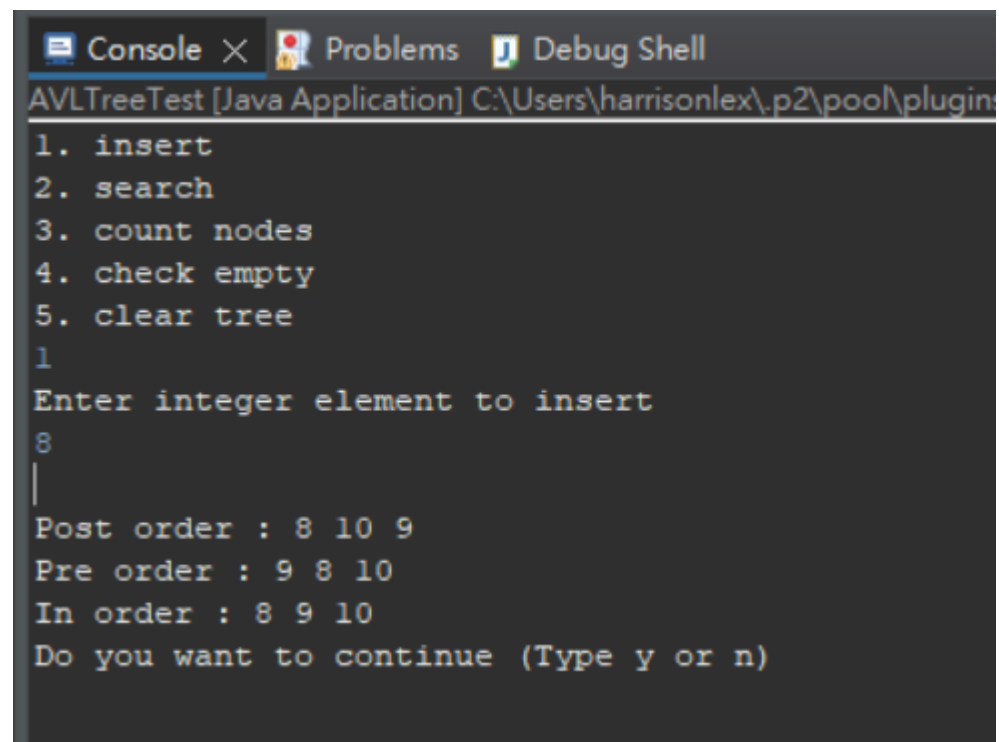
Name	Value
rotateWithLeftChild() returnec	AVLNode (id=25)
this	AVLTree (id=21)
x	8
t	AVLNode (id=25)
data	8
height	2
left	AVLNode (id=24)
data	9
height	1
left	AVLNode (id=25)
data	8
height	2
left	AVLNode (id=24)
right	null
right	null

所以我再用 F5 到 rotateWithLeftChild() 這個 function 裡看，確認錯誤是在 77 行執行 rotateWithLeftChild() 後，我對程式碼做修改如下

```
76         if( x < t.left.data )
77             t = rotateWithLeftChild( t );
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98     /* Rotate binary tree node with left child */
99     private AVLNode rotateWithLeftChild(AVLNode k2)
100     {
101         AVLNode k1 = k2.left;
102         k2.left = k1.right;
103         k1.right = k2;
104         k2.height = max( height( k2.left ), height( k2.right ) ) + 1;
105         k1.height = max( height( k1.left ), k2.height ) + 1;
106         return k1;
107     }
```

原來是 77 行參數放錯，應該要放入節點本身而非其左子樹，還有 102 行和 103 行的 left 和 right 寫反了

修正後就可正常輸出



The screenshot shows a Java IDE's console window with the title bar 'Console', 'Problems', and 'Debug Shell'. The active tab is 'Console', showing the output of an 'AVLTreeTest' application. The application prompts the user to select from five menu options: 1. insert, 2. search, 3. count nodes, 4. check empty, and 5. clear tree. Option 1 is selected, and the user is prompted to 'Enter integer element to insert'. The user enters '8'. The application then displays the traversal results: 'Post order : 8 10 9', 'Pre order : 9 8 10', and 'In order : 8 9 10'. Finally, it asks 'Do you want to continue (Type y or n)'.

```
AVLTreeTest [Java Application] C:\Users\harrisonlex\p2\pool\plugins
1. insert
2. search
3. count nodes
4. check empty
5. clear tree
1
Enter integer element to insert
8
|
Post order : 8 10 9
Pre order : 9 8 10
In order : 8 9 10
Do you want to continue (Type y or n)
```

剛剛的 10, 9, 8 插入屬於右旋，為測試左旋我插入 10, 11, 12 發現同樣 bug，直接比照第一個 bug 找到錯誤處如下

```
84         if( height( t.right ) - height( t.left ) == 2 )
85             if( x > t.right.data)
86                 t = rotateWithRightChild( t.right );|
87             else
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109     /* Rotate binary tree node with right child */
110     private AVLNode rotateWithRightChild(AVLNode k1)
111     {
112         AVLNode k2 = k1.right;
113         k1.left = k2.right;
114         k2.right = k1;
115         k1.height = max( height( k1.left ), height( k1.right ) ) + 1;
116         k2.height = max( height( k2.right ), k1.height ) + 1;
117         return k2;
118     }
119     /**
```

然後修改如下

```
85         if( x > t.right.data)
86             t = rotateWithRightChild( t );
87         else
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109     /* Rotate binary tree node with right child */
110     private AVLNode rotateWithRightChild(AVLNode k1)
111     {
112         AVLNode k2 = k1.right;
113         k1.right = k2.left;
114         k2.left = k1;
115         k1.height = max( height( k1.left ), height( k1.right ) ) + 1;
116         k2.height = max( height( k2.right ), k1.height ) + 1;
117         return k2;
118     }
```

至此就兩個 bug 都找到並修正，可以進行表一的所規定的所有互動