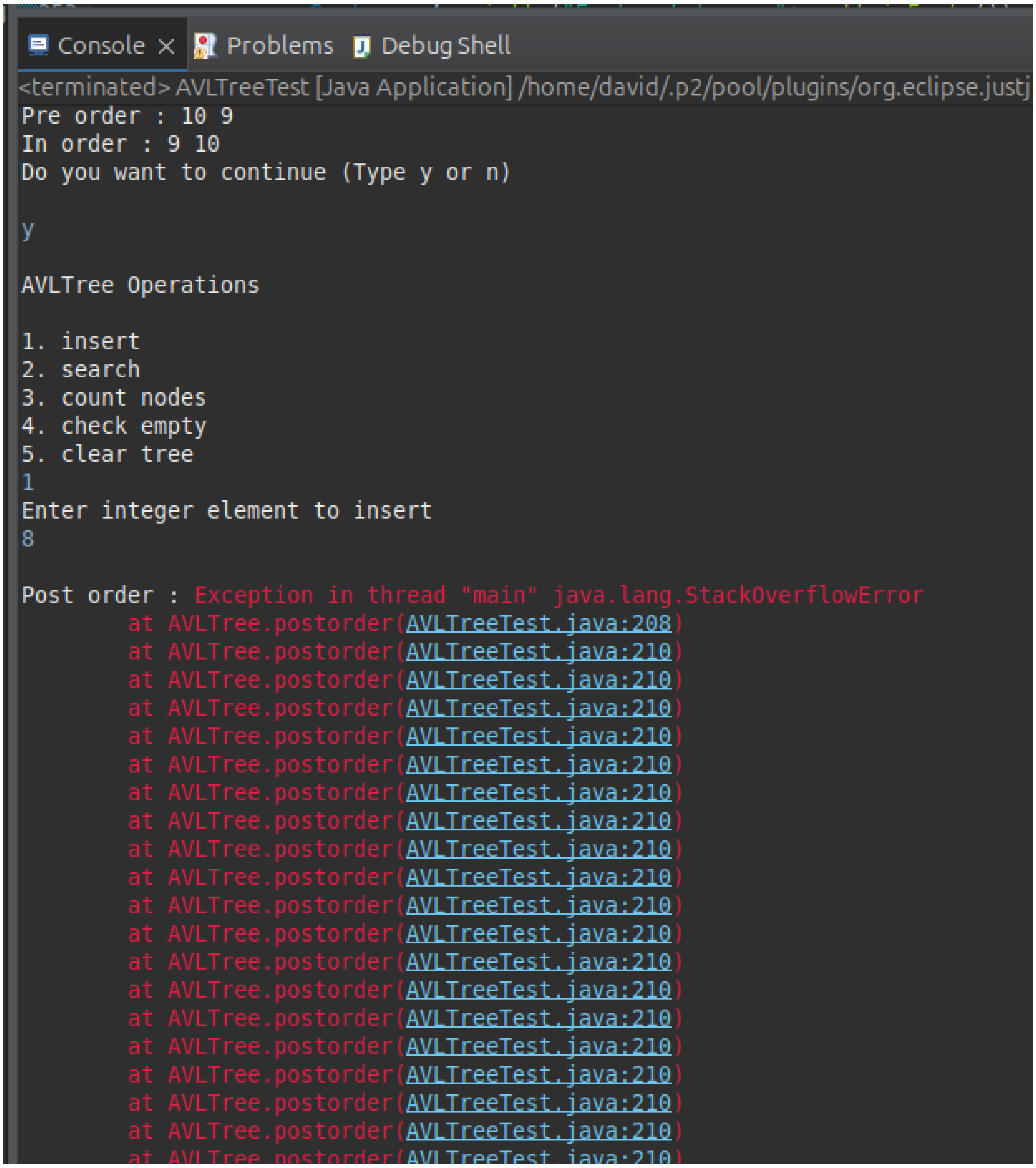
HW1: debug an AVL tree

> 軟工碩一 113525011 江大衛
> 題目要求

ANS:

bug1

- 原因：在輸入 8 的時候會發生錯誤
- 發生：不斷遞迴導致 stack overflow error.

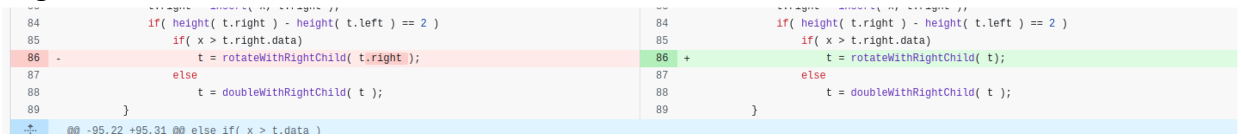


```
<terminated> AVLTreeTest [Java Application] /home/david/.p2/pool/plugins/org.eclipse.justj
Pre order : 10 9
In order : 9 10
Do you want to continue (Type y or n)

y

AVLTree Operations

1. insert
2. search
3. count nodes
4. check empty
5. clear tree
1
Enter integer element to insert
8

Post order : Exception in thread "main" java.lang.StackOverflowError
        at AVLTree.postorder(AVLTreeTest.java:208)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
        at AVLTree.postorder(AVLTreeTest.java:210)
```

PROF

修復

- bug1: `rotateWithLeftChild` 傳送的參數不正確

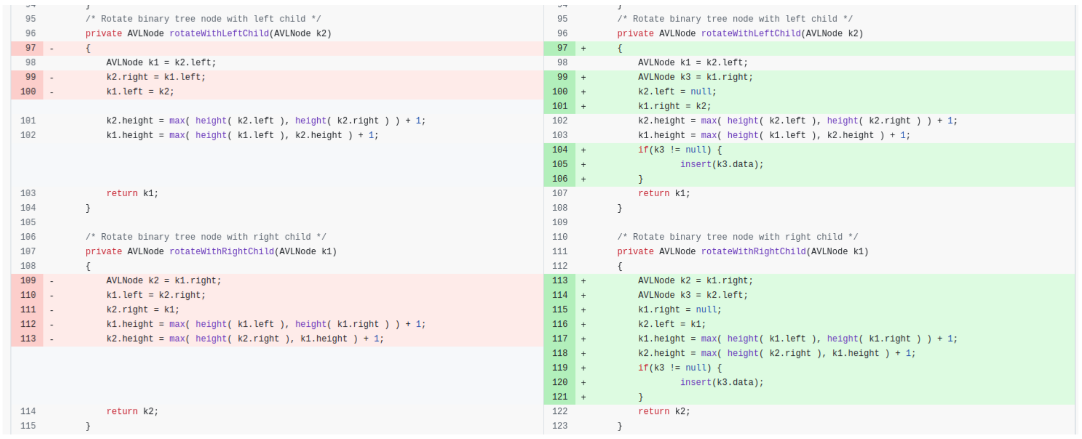


```
        if( height( t.right ) - height( t.left ) == 2 )
            if( x > t.right.data )
-               t = rotateWithRightChild( t.right );
            else
                t = doubleWithRightChild( t );
    }
```

```
        if( height( t.right ) - height( t.left ) == 2 )
            if( x > t.right.data )
+               t = rotateWithRightChild( t );
            else
                t = doubleWithRightChild( t );
    }
```

```
@@ -95,22 +95,31 @@ else if( x > t.data )
```

- bug2: `rotateWithLeftChild`, `rotateWithRightChild rotate 錯誤
  - 解法
    - 新增一個 AVLNode 做為回傳值。
    - 將 function 傳近來的參數進行旋轉，並讓新增的 AVLNode 做為 root
    - 在旋轉的過程中，如果有資料會被排擠到，則將它重新做一次 insert.



```
    /* Rotate binary tree node with left child */
    private AVLNode rotateWithLeftChild(AVLNode k2)
-   {
        AVLNode k1 = k2.left;
-       k2.right = k1.left;
-       k1.left = k2;

        k2.height = max( height( k2.left ), height( k2.right ) ) + 1;
        k1.height = max( height( k1.left ), k2.height ) + 1;



        return k1;
    }

    /* Rotate binary tree node with right child */
    private AVLNode rotateWithRightChild(AVLNode k1)
    {
-       AVLNode k2 = k1.right;
-       k1.left = k2.right;
-       k2.right = k1;
-       k1.height = max( height( k1.left ), height( k1.right ) ) + 1;
-       k2.height = max( height( k2.right ), k1.height ) + 1;




        return k2;
    }
```

```
    /* Rotate binary tree node with left child */
    private AVLNode rotateWithLeftChild(AVLNode k2)
+   {
        AVLNode k1 = k2.left;
+       AVLNode k3 = k1.right;
+       k2.left = null;
+       k1.right = k2;
        k2.height = max( height( k2.left ), height( k2.right ) ) + 1;
        k1.height = max( height( k1.left ), k2.height ) + 1;
+       if(k3 != null) {
+               insert(k3.data);
+       }
        return k1;
    }

    /* Rotate binary tree node with right child */
    private AVLNode rotateWithRightChild(AVLNode k1)
    {
+       AVLNode k2 = k1.right;
+       AVLNode k3 = k2.left;
+       k1.right = null;
+       k2.left = k1;
+       k1.height = max( height( k1.left ), height( k1.right ) ) + 1;
+       k2.height = max( height( k2.right ), k1.height ) + 1;
+       if(k3 != null) {
+               insert(k3.data);
+       }
        return k2;
    }
```