

軟體工程

HW1:Debugging

112522081 戴芷柔

結論：找出的 bug

Bug 1：選擇 rotate 的節點錯誤

Bug 2：rotate 程式碼邏輯錯誤，造成無窮迴圈

以下為找出 bug 過程

先以作業給予的以下提示測試資料

並在 INSERT 8 時出錯

```
AVLTree Tree Test
```

```
AVLTree Operations
```

1. insert
2. search
3. count nodes
4. check empty
5. clear tree

```
4
```

```
Empty status = true
```

```
Post order :
```

```
Pre order :
```

```
In order :
```

```
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

10

Post order : 10

Pre order : 10

In order : 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

9

Post order : 9 10

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 :

Error!

發現錯誤後先將斷點設在每次輸入完一個 action 後

觀察 variables 狀況

輸入完整 1:10

Name	Value
no method return value	
args	String[0] (id=20)
scan	Scanner (id=22)
avlt	AVLTree (id=28)
root	AVLNode (id=34)
data	10
height	0
left	null
right	null

輸入完整 1:9

Name	Value
no method return value	
args	String[0] (id=20)
scan	Scanner (id=22)
avlt	AVLTree (id=28)
root	AVLNode (id=34)
data	10
height	1
left	AVLNode (id=51)
data	9
height	0
left	null
right	null
right	null

在輸入完 1:8 後，執行到 `avlt.postorder()` 造成程式錯誤

並顯示在執行到 210 行 postorder 時產生錯誤

```
206 private void postorder(AVLNode r)
207 {
208     if (r != null)
209     {
210         postorder(r.left);
211         postorder(r.right);
212         System.out.print(r.data + " ");
213     }
214 }
```

觀察上述用到的參數 r 可看出 node 9 跟 node 8 不尋常地出現了多次
貌似出現了無窮迴圈，導致無法輸出 node

[illegible]

因此重新審視在 node 8 添加時，問題出在哪

先檢查添加 node 8 之前的 AVLtree 是否正確，如下圖(正確)

Name	Value
no method return value	
▼ this	AVLTree (id=20)
▼ root	AVLNode (id=38)
data	10
height	1
▼ left	AVLNode (id=63)
data	9
height	0
left	null
right	null
right	null
data	8

嘗試將程式中斷在已經添加 node 8 但尚未平衡之前

執行到加完 node 8 並在尚未進行平衡之前，無窮迴圈還沒出現，

Name	Value
no method return value	
> this	AVLTree (id=28)
x	8
▼ t	AVLNode (id=30)
data	10
height	1
▼ left	AVLNode (id=40)
data	9
height	1
▼ left	AVLNode (id=47)
data	8
height	0
left	null
right	null
right	null
right	null

接著觀察 rotateWithLeftChild

進入 rotateWithLeftChild 發現旋轉要用的節點錯誤 (bug 1)

此為 LL 型，邏輯判斷這邊在旋轉時應該是使用 node 10，
因此在前面判斷選擇要 rotate 的節點有 bug

Name	Value
no method return value	
> this	AVLTree (id=21)
▼ k2	AVLNode (id=24)
data	9
height	1
▼ left	AVLNode (id=25)
data	8
height	0
left	null
right	null
right	null

找到呼叫 rotate 的原始程式：

```
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     }
```

更改程式：

```
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 );
78             else
79                 t = doubleWithLeftChild( t );
80     }
```

更改完之後進入 rotateWithLeftChild 的參數已更正

Name	Value
no method return value	
> this	AVLTree (id=21)
▼ k2	AVLNode (id=22)
data	10
height	1
▼ left	AVLNode (id=24)
data	9
height	1
> left	AVLNode (id=25)
right	null
right	null

接著需要解決另一個無線迴圈的 bug (bug 2)

在觀察 rotateWithLeftChild 程式後

簡單畫圖後發現這根本畫不出來

有邏輯錯誤

因此需要修改程式

```

97● private AVLNode rotateWithLeftChild(AVLNode k2)
98   {
99       AVLNode k1 = k2.left;
100       k2.right = k1.left;
101       k1.left = k2;
102       k2.height = max( height( k2.left ), height( k2.right ) ) + 1;
103       k1.height = max( height( k1.left ), k2.height ) + 1;
104       return k1;
105   }

```

更改後的程式：

```

98● private AVLNode rotateWithLeftChild(AVLNode k2)
99   {
100       AVLNode k1 = k2.left;
101       k2.left = k1.right;
102       k1.right = k2;
103       k2.height = max( height( k2.left ), height( k2.right ) ) + 1;
104       k1.height = max( height( k1.left ), k2.height ) + 1;
105       return k1;
106   }

```

根據這個測試經驗

添加其他測試：（測試另一個旋轉平衡-rotateWithRightChild）

AVLTree Tree Test

AVLTree Operations

1. insert
2. search
3. count nodes
4. check empty
5. clear tree

1

Enter integer element to insert

8

Post order : 10

Pre order : 10

In order : 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

9

Post order : 8 10

Pre order : 10 8

In order : 8 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

10

Post order :

Error!

在 rotateWithRightChild 得到同樣錯誤

Name	Value
max() returned	1
> this	AVLTree (id=21)
▼ k1	AVLNode (id=24)
data	9
height	1
left	null
▼ right	AVLNode (id=27)
data	10
height	2
left	null
▼ right	AVLNode (id=24)
data	9
height	1
left	null
▼ right	AVLNode (id=27)
data	10
height	2
left	null
▼ right	AVLNode (id=24)
data	9
height	1
left	null
▼ right	AVLNode (id=27)
data	10
height	2
left	null

因此同樣在選擇節點跟旋轉平衡邏輯進行調整

rotateWithRightChild 原始程式：

```
109● private AVLNode rotateWithRightChild(AVLNode k1)
110     {
111         AVLNode k2 = k1.right;
112         k1.left = k2.right;
113         k2.right = k1;
114         k1.height = max( height( k1.left ), height( k1.right ) ) + 1;
115         k2.height = max( height( k2.right ), k1.height ) + 1;
116         return k2;
117     }
```

修改過後：

```
109● private AVLNode rotateWithRightChild(AVLNode k1)
110     {
111         AVLNode k2 = k1.right;
112         k1.right = k2.left;
113         k2.left = k1;
114         k1.height = max( height( k1.left ), height( k1.right ) ) + 1;
115         k2.height = max( height( k2.right ), k1.height ) + 1;
116         return k2;
117     }
```

選擇節點原始程式：

```
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         }
```

更改程式：

```
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 );
87             else
88                 t = doubleWithRightChild( t );
89         }
```

經過修改過後的程式，可通過原本的測試資料。