

## HW: xUnit Test goes deep

Write the **complete** tests for an AVL implementation

說明:

在 lab練習中, 你已經拿到了 AVL tree 的java 程式碼。這個程式是一個 AVL tree 的實作, 請撰寫一些簡單的測試案例。

只需對 public method 測試即可, 這個AVL程式的 public method 有:

- AVLtree()
- isEmpty()
- makeEmpty()
- insert()
- countNodes
- search()
- inorder()
- preorder()
- postorder()

我幫你 google 了一下, AVL tree 的規格:

1. In an AVL tree, the [heights](#) of the two [child](#) subtrees of any node differ by at most one; if at any time they differ by more than one, rebalancing is done to restore this property.
2. Lookup, insertion, and deletion all take  $O(\log n)$  time in both the average and worst cases, where  $n$  is the number of nodes in the tree prior to the operation

你的任務:

1. 請你設計一套 unit test cases 他能夠完整地 guard 上述的規格
2. 所謂的完整地 unit test cases 按照老師的投影片解說, 每個 test cases 的設計你要能夠說出個所以然

- A. Code coverage: 同學不需使用線上工具測 Coverage, 請在註解中分析自己寫的test的coverage即可
- B. Partition testing
- C. Boundary tests
- D. Negative tests
- E. Performance tests: 同學可以創建 Node 數量超大的 AVL tree, 看 insert或delete有沒有滿足 $O(\log n)$

你所繳交的內容與規定:

1. 一個 test class 名稱為 AVLtreeTest.java
2. 為你的 test method 名字做一個專業的命名, 命名不佳助教會酌以扣分  
<https://osherove.com/blog/2005/4/3/naming-standards-for-unit-tests.html>  
 。  
 下面這個連結是另外一個 test method naming conventions。  
<https://enterprisecraftsmanship.com/posts/you-naming-tests-wrong/>  
 這個 naming convention 把 MUT 名字放在前面。但是有人認為這不好, 因為class method 名字可能未來會改變。
3. 一群 test methods。每一個method 的名字除了都按照上述的link 來進行命名之外, 在尾巴請加入都要非常明確地讓助教知道你是根據哪個思維來設計這個test case。例如:
  - A. make\_rotation\_givenXXXX\_expectedforXXXX\_byCoverage: 你在測試 AVL的旋轉, 主軸是 Coverage
  - B. test\_balance\_givenXXXX\_expectedforXXXX\_byPartition: 你在測試 balance 主軸是 partition testing
 當然囉, 加上這個 byCoverage 只是 for this course only。真正的專業你不需要加上這些東西
4. 在 test methods 裡面寫註解說明你想測什麼, 讓助教來打分數。下圖是一個範例, 也算是單元測試的正規註解 //give //when //then。你必須要在這些註解後面寫上更多的測試意圖。例如你的測試方法名稱已經是 make\_rotation\_byCoverage。我們已經清楚你的測試是為了達成某塊程式碼的 coverage。但是在註解後面, 我們要看到更進一步的說明與解釋。例如為何你準備的 //given //when 可以觸動 rotation。也許你必須用文字畫一個簡單的 binary tree 來做說明。
5. 請大家將所有test集中一個檔案, 並將註解寫清楚, 助教比較好送同學分數!!

```
@Test
void itShouldAddTwoNumbers() {
    // given
    int numberOne = 20;
    int numberTwo = 30;

    // when
    int result = underTest.add(numberOne, numberTwo);

    // then
    int expected = 50;
    assertThat(result).isEqualTo(expected);
}
```

**HINT and Suggestion:** 為了達成你的目標, 你可以假設 AVLtree.java 其實也是你自己寫的。所以如果你必須 add and change some code in CUT for testing purpose, it is perfectly fine.