Day17. LeetCode我這不是來了嗎?－LeetCode\_First Missing Positive

Codewars的介面實在是用的很膩(被揍爆

今天來改用LeetCode挑一題唄XD

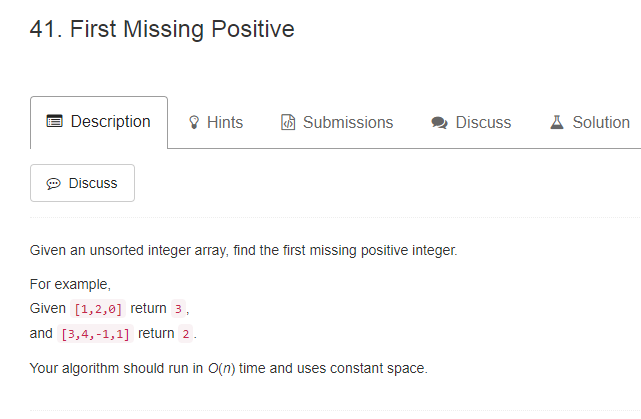


以前我實在是不太相信LeetCode裡面的題目真的可以一下就寫得很漂亮的….

自從我碰到了TDD我考試就開始考一百分了呢(被揍

或許是自己以前不會拆解分析題目吧，所以每次寫LeetCode都沒辦法過Runtime超煩der

廢話我就不多說惹，今天的題目長這樣



題目給的需求鰻不明確的，跟我們的使用者一樣(被揍爆

不過我猜他如果是輸入0以下的數值時就是要找到1

如果是負數到正數之間，0是空著的話也不會輸出0而是1，目前暫且先這個樣子。

現在就來拆一下題目吧。

1. 找出陣列中最大的值->Linq的Max方法
2. 找出是否有相同的元素->Linq的Contains方法
3. 找出來的元素必須要大於0

現在就來想一下最簡單的測試案例吧!

就先從空的陣列開始做輸入吧!

```

[TestMethod]

public void Input\_null\_Should\_Be\_1()

{

Assert.AreEqual(1,Solution.FirstMissingPositive(new int[]{}));

}

```

而Production Code 也就是老樣子會長成這個樣子

```

public static int FirstMissingPositive(int[] nums)

{

throw new **NotImplementedException**();

}  
```

老樣子，跑個測試，沒過很正常，紅燈，commit一下

接下來把Production Code改一下，用最簡單的方式解決他!

```

public static int FirstMissingPositive(int[] nums)

{

if (nums.Length < 1)

{

return 1;

}

return 1;

}  
```

接下來跑個測試，PASS!

接下來寫一個輸入null過了，就來寫一下輸入只有0的時候呢

```

[TestMethod]

public void Input\_0\_Should\_Be\_1()

{

Assert.AreEqual(1,Solution.FirstMissingPositive(new int[]{0}));

}

```

一樣是過了，因為原本的Production Code涵蓋了輸入0的需求。

所以我們現在來輸入1。

```

[TestMethod]

public void Input\_1\_Should\_Be\_2()

{

Assert.AreEqual(2, Solution.FirstMissingPositive(new int[] { 1 }));

}

```

接下來跑一下測試吧，紅燈!

然後改個Production Code

```

public static int FirstMissingPositive(int[] nums)

{

if (nums.Length < 1)

{

return 1;

}

return nums[0] + 1;

}

```

寫完就來跑個測試，PASS，Commit一下唄!

再來寫一個輸入負號的測試。

```

[TestMethod]

public void Input\_f1\_Should\_Be\_1()

{

Assert.AreEqual(1, Solution.FirstMissingPositive(new int[] { -1 }));

}

```

接下來就要改Production Code了

這邊就要考慮到輸出一定要大於0而且要從1開始判斷，所以for迴圈可以從1開始，從陣列最大值+1結束，並且要加入Contains方法了。

原本最後一行的return num[0]+1也要改成1才能符合只輸入負號的測試。

```

if (nums.Length < 1)

{

return 1;

}

for (int i = 1; i <= nums.Max() + 1; i++)

{

if (!nums.Contains(i))

{

return i;

}

}

return 1;

```

基本上目前的程式碼已經可以涵蓋到所有範圍，所以我們就多加幾個測試案例吧!

以下是今天所有的測試程式碼

```

[TestClass]

public class UnitTest1

{

[TestMethod]

public void Input\_null\_Should\_Be\_1()

{

Assert.AreEqual(1, Solution.FirstMissingPositive(new int[] { }));

}

[TestMethod]

public void Input\_0\_Should\_Be\_1()

{

Assert.AreEqual(1, Solution.FirstMissingPositive(new int[] { 0 }));

}

[TestMethod]

public void Input\_1\_Should\_Be\_2()

{

Assert.AreEqual(2, Solution.FirstMissingPositive(new int[] { 1 }));

}

[TestMethod]

public void Input\_f1\_Should\_Be\_1()

{

Assert.AreEqual(1, Solution.FirstMissingPositive(new int[] { -1 }));

}

[TestMethod]

public void Input\_f2f3and5and6\_Should\_Be\_1()

{

Assert.AreEqual(1, Solution.FirstMissingPositive(new int[] { -2, -3, 5, 6 }));

}

[TestMethod]

public void Input\_1n2n3n4\_Should\_Be\_5()

{

Assert.AreEqual(5, Solution.FirstMissingPositive(new int[] { 1, 2, 3, 4 }));

}

[TestMethod]

public void Input\_1n6n5()

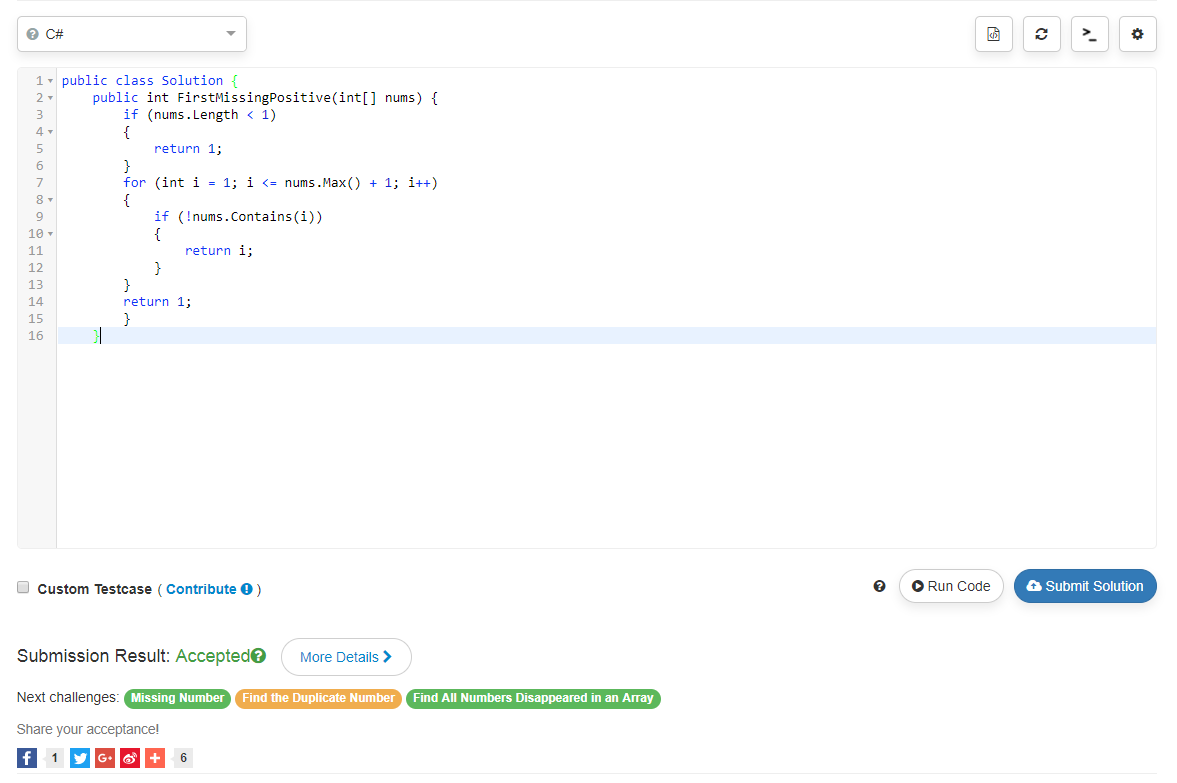
{

Assert.AreEqual(2, Solution.FirstMissingPositive(new int[] { 1, 6, 5 }));}

}

```

然後在LeetCode上提交! Pass!!



通過! PASS!

LeetCode我記得以前可以看到別人的Code啊..

為毛我今天就找不到勒..

有時間我再研究一下好惹

今天的TDD練習就到這裡

LeetCode也是可以用TDD寫的嘛!!!(被揍

Git url :

<https://github.com/SQZ777/LeetCode_ReverseInteger>

下一題，明天見!