hw7 p1 answer

November 25, 2018

1 Assignment 7 part1

1.0.1 MACS 30000, Dr. Evans

1.0.2 Dongcheng Yang

1.0.3 Nov. 20

1.0.4 1. Unit Testing in Python.

problem1

In [1]:
```
# original function
def smallest_factor(n):
    """Return the smallest prime factor of the positive integer n."""
    if n == 1:
        return 1
    for i in range(2, int(n**.5)):
        if n % i == 0:
            return i
    return n
```

In [2]:
```
import p1a

def test_smallest_factor1():
    assert p1a.smallest_factor(9) == 3, "failed on full square"

def test_smallest_factor2():
    assert p1a.smallest_factor(6) == 2, "failed on small numbers"
```

In [12]:
```
!py.test test_p1a.py
```

```
============================= test session starts =============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0
rootdir: C:\Users\pcc\Desktop\persp-analysis_A18-master\Assignments\A7\problem1, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, cov-2.6.0, arraydiff-0.2
collected 2 items

test_p1a.py FF                                                           [100%]
```

```
================================= FAILURES =================================
___________________________ test_smallest_factor1 ___________________________

    def test_smallest_factor1():
>       assert p1a.smallest_factor(9) == 3, "failed on full square"
E       AssertionError: failed on full square
E       assert 9 == 3
E        +  where 9 = <function smallest_factor at 0x000000C959506A60>(9)
E        +    where <function smallest_factor at 0x000000C959506A60> = p1a.smallest_factor

test_p1a.py:10: AssertionError
___________________________ test_smallest_factor2 ___________________________

    def test_smallest_factor2():
>       assert p1a.smallest_factor(6) == 2, "failed on small numbers"
E       AssertionError: failed on small numbers
E       assert 6 == 2
E        +  where 6 = <function smallest_factor at 0x000000C959506A60>(6)
E        +    where <function smallest_factor at 0x000000C959506A60> = p1a.smallest_factor

test_p1a.py:13: AssertionError
========================== 2 failed in 0.29 seconds ==========================
```

Two special cases are included in the pytest. The first one is the square of a prime number and the second one is a small number. The small number case failed because the range (2, int(n**.5)) was not able to include anything.

In [17]:
```
# corrected version

        def smallest_factor(n):
            """Return the smallest prime factor of the positive integer n."""
            if n == 1: return 1
            for i in range(2, int(n**.5)+1):
                if n % i == 0: return i
            return n

        # This corrected version could pass all tests cases with full coverage
```

problem2 Here is the code to check my coverage of the function smallest_factor() from problem1.
The corrected version of the function smallest_factor() is saved in file "p1a.py "
and test cases in "test_p1a.py". Then I ran "py.test --cov" to test.

In [16]:
```
!py.test --cov
```

```
============================= test session starts =============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0
```

```
rootdir: C:\Users\pcc\Desktop\persp-analysis_A18-master\Assignments\A7\problem1, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, cov-2.6.0, arraydiff-0.2
collected 2 items

test_p1a.py .                                                            [ 50%]
test_p1b.py .                                                            [100%]

----------- coverage: platform win32, python 3.6.5-final-0 -----------
Name          Stmts   Miss  Cover
---------------------------------
p1a.py            6      0   100%
p1b.py           11      0   100%
test_p1a.py      10      0   100%
test_p1b.py      11      0   100%
---------------------------------
TOTAL            38      0   100%


========================== 2 passed in 0.22 seconds ==========================
```

The coverage for both files are 100%.

Here is the code to check the function month_length(). The original function is stored in the file "p1b.py" and the test cases are saved in the file "test_p1b.py". Then I ran "py.test --cov" to test. The coverage is also 100%.

```
In [ ]: #original function

        def month_length(month, leap_year=False):
            """Return the number of days in the given month."""
            if month in {"September", "April", "June", "November"}:
                return 30
            elif month in {"January", "March", "May", "July","August", "October", "December"}:
                return 31
            if month == "February":
                if not leap_year:
                    return 28
                else:
                    return 29
            else:
                return None

In [18]: import p1b
         def test_month_length():
             assert p1b.month_length("January") == 31
             assert p1b.month_length("February") == 28
             assert p1b.month_length("February",leap_year=True) == 29
             assert p1b.month_length("July") == 31
```

```
            assert p1b.month_length("August") == 31
            assert p1b.month_length("September") == 30
            assert p1b.month_length("October") == 31
            assert p1b.month_length("else") == None
```

```
In [19]: !py.test --cov
```

```
============================= test session starts ============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0
rootdir: C:\Users\pcc\Desktop\persp-analysis_A18-master\Assignments\A7\problem1, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, cov-2.6.0, arraydiff-0.2
collected 2 items

test_p1a.py .                                                            [ 50%]
test_p1b.py .                                                            [100%]

---------- coverage: platform win32, python 3.6.5-final-0 ----------
Name          Stmts   Miss  Cover
---------------------------------
p1a.py            6      0   100%
p1b.py           11      0   100%
test_p1a.py      10      0   100%
test_p1b.py      11      0   100%
---------------------------------
TOTAL            38      0   100%


========================== 2 passed in 0.20 seconds ==========================
```

problem3 Here is the code to check the function operate(). The original function is stored in the file "p1c.py" and the test cases are saved in the file "test_p1c.py". Then I ran "py.test --cov" to test.

```
In [21]: #original function

         def operate(a, b, oper):
             """Apply an arithmetic operation to a and b."""
             if type(oper) is not str:
                 raise TypeError("oper must be a string")
             elif oper == '+':
                 return a + b
             elif oper == '-':
                 return a - b
             elif oper == '*':
                 return a * b
             elif oper == '/':
                 if b == 0:
```

```
                raise ZeroDivisionError("division by zero is undefined")
            return a / b
        raise ValueError("oper must be one of '+', '/', '-', or '*'")
```

```
In [24]: import p1c
         import pytest

         def test_cases():
             assert p1c.operate(1,2,"+") == 3
             assert p1c.operate(1,2,"-") == -1
             assert p1c.operate(1,2,"*") == 2
             assert p1c.operate(1,2,"/") == 0.5
             with pytest.raises(ZeroDivisionError) as err:
                 p1c.operate(1,0,'/')
             assert err.value.args[0]=="division by zero is undefined"

             with pytest.raises(TypeError) as err2:
                 p1c.operate(1,0,0)
             assert err2.value.args[0]=="oper must be a string"

             with pytest.raises(ValueError) as err3:
                 p1c.operate(1,0,'<')
             assert err3.value.args[0]=="oper must be one of '+', '/', '-', or '*'"
```

```
In [25]: !py.test --cov

============================= test session starts ==============================
platform win32 -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0
rootdir: C:\Users\pcc\Desktop\persp-analysis_A18-master\Assignments\A7\problem1, inifile:
plugins: remotedata-0.2.1, openfiles-0.3.0, doctestplus-0.1.3, cov-2.6.0, arraydiff-0.2
collected 3 items

test_p1a.py .                                                            [ 33%]
test_p1b.py .                                                            [ 66%]
test_p1c.py .                                                            [100%]

----------- coverage: platform win32, python 3.6.5-final-0 -----------
Name          Stmts   Miss  Cover
---------------------------------
p1a.py            6      0   100%
p1b.py           11      0   100%
p1c.py           15      0   100%
test_p1a.py      10      0   100%
test_p1b.py      11      0   100%
test_p1c.py      17      0   100%
---------------------------------
TOTAL            70      0   100%
```

```
========================== 3 passed in 0.26 seconds ==========================
```