

09 Exercises

September 13, 2020

0.1 Exercise 09.1 (checking data validity)

The Fibonacci series is valid only for $n \geq 0$. Add to the Fibonacci function in this notebook a check that raises an exception if $n < 0$. Try some invalid data cases to check that an exception is raised.

Optional: Use `pytest` to test that an exception *is* raised for some $n < 0$ cases.

```
[0]: def f(n):  
    "Compute the nth Fibonacci number using recursion"  
    if n == 0:  
        return n  
    elif n == 1:  
        return n  
    elif n < 0:  
        raise ValueError("Value must be a positive integer")  
    else:  
        return f(n-1) + f(n-2)  
    raise NotImplementedError()
```

```
[0]: # Perform some tests  
assert f(0) == 0  
assert f(1) == 1  
assert f(2) == 1  
assert f(3) == 2  
assert f(10) == 55  
assert f(15) == 610  
  
# Check that ValueError is raised for n < 0  
import pytest  
with pytest.raises(ValueError):  
    f(-1)  
with pytest.raises(ValueError):  
    f(-2)
```

0.2 Exercise 09.2 (raising exceptions)

Modify your program from the bisection exercise in Activity 04 to raise an error if the maximum number of iterations is exceeded. Reduce the maximum allowed iterations to test that an exception

is raised.

Add any other checks on the input data that you think are appropriate.

```
[0]: def my_f(x):  
    "Evaluate polynomial function"  
    return x**3 - 6*x**2 + 4*x + 12  
  
def compute_root(f, x0, x1, tol, max_it, it_count=0):  
    "Compute roots of a function using bisection"  
    it_count += 1  
    # Compute midpoint  
    x_mid = (x0 + x1)/2  
    f = my_f(x0)  
    f_mid = my_f(x_mid)  
    if abs(f_mid) > tol:  
        # Condition:  
        if f*f_mid < 0:  
            x1=x_mid  
        else:  
            x0=x_mid  
        return compute_root(f, x0, x1, tol, max_it, it_count)  
    elif it_count > max_it:  
        raise RuntimeError("Oops,maximum number of iterations is exceeded.")  
    else:  
        return x_mid, f_mid, it_count
```

```
[0]: # Test with max_it = 30  
x, f, num_it = compute_root(my_f, x0=3, x1=6, tol=1.0e-6, max_it=30)  
  
# Test with max_it = 20  
with pytest.raises(RuntimeError):  
    x, f, num_it = compute_root(my_f, x0=3, x1=6, tol=1.0e-6, max_it=20)
```