Homework 5

Exercise 1

表示有理数的类

题目：

有理数的一般形式是 a/b, 其中 a 是整数, b 是正整数, 并且当 a 非 0 时 $|a|$ 和 b 的最大公约数是1。实现`Rational`类表示有理数和其运算。5.7已列出了部分代码, 需要实现标注了`“to be implemented”`的函数。`Rational`类的属性`nu`和`de`分别表示分子和分母。函数`__add__`、`_sub__`、`__mul_`和`__truediv__`分别进行加减乘除运算, 然后返回一个新创建的`Rational`对象作为运算结果。函数`__eq__`、`__ne__`、`__gt__`、`__lt__`别是: `==`、`!=`、`>`、`<`、`>=`、`<=`。例如表达式`Rational(6,-19) > Rational(14,-41)`在求值时被转换成方法调用`Rational(6,-19)__gt__(Rational(14,-41))`。函数`test`测试这些函数。 `gcd`函数要求形参`a`和`b`都是正整数, 如果其中出现0或负数, 递归不会终止。

源代码：

```
print("This is a python code for homework5-1: Rational Number Class")
def gcd(a, b):
    if a == b:
        return a
    elif a > b:
        return gcd(a-b, b)
    else:
        return gcd(a, b-a)


class Rational:
    def __init__(self, n=0, d=1):
        if n != 0:
            fgcd = gcd(abs(n), abs(d))
        else:
            fgcd = 1
        n = n/fgcd
        d = d/fgcd
        if d < 0:
            n = -n
            d = -d
        if n == 0:
            d = 1
        _nu = n
        _de = d
        self.__dict__['nu'] = _nu
        self.__dict__['de'] = _de

    def __setattr__(self, name, value):
```

```
        raise TypeError('Error: Rational objects anu demuntable')

    def __str__(self):
        return '%d/%d' % (self.nu, self.de)

    def __add__(self, other):
        return Rational(self.nu*other.de+other.nu*self.de, self.de*other.de)

    def __sub__(self, other):
        return Rational(self.nu*other.de-other.nu*self.de, self.de*other.de)

    def __mul__(self, other):
        return Rational(self.nu*other.nu, self.de*other.de)

    def __truediv__(self, other):
        return Rational(self.nu*other.de, self.de*other.nu)

    def __eq__(self, other):
        temp = self - other
        if temp.nu == 0:
            return True
        else:
            return False

    def __ne__(self, other):
        temp = self - other
        if temp.nu != 0:
            return True
        else:
            return False

    def __gt__(self, other):
        temp = self - other
        if temp.nu > 0:
            return True
        else:
            return False

    def __lt__(self, other):
        temp = self - other
        if temp.nu < 0:
            return True
        else:
            return False

    def __ge__(self, other):
        temp = self - other
        if temp.nu >= 0:
            return True
```

```
        else:
            return False

    def __le__(self, other):
        temp = self - other
        if temp.nu <= 0:
            return True
        else:
            return False

def test():
    testsuite = [
        ('Rational(2, 3) + Rational(-70, 40)', Rational(-13, 12)),
        ('Rational(-20, 3) - Rational(120, 470)', Rational(-976, 141)),
        ('Rational(-6, 19) * Rational(-114, 18)', Rational(2, 1)),
        ('Rational(-6, 19) / Rational(-114, -28)', Rational(-28, 361)),

        ('Rational(-6, 19) == Rational(-14, 41)', False),
        ('Rational(-6, 19) != Rational(-14, 41)', True),
        ('Rational(6, -19) > Rational(14, -41)', True),
        ('Rational(-6, 19) < Rational(-14, 41)', False),
        ('Rational(-6, 19) >= Rational(-14, 41)', True),
        ('Rational(6, -19) <= Rational(14, -41)', False),
        ('Rational(-15, 8) == Rational(120, -64)', True),
    ]
    for t in testsuite:
        try:
            result = eval(t[0])
        except:
            print('Error in evaluating ' + t[0]); continue

        if result != t[1]:
            print('Error:  %s != %s' % (t[0], t[1]))

if __name__ == '__main__':
    test()
```

运行结果：

```
/usr/local/bin/python3.9 "/Users/wangyijie/Library/Mobile
Documents/com~apple~CloudDocs/Study_in_USTC/杂事/python科学计算/hw5/hw5-2.py"

This is a python code for homework5-1: Integrator Class

Process finished with exit code 0
```

可见，测试函数没有报错，说明程序运行正常。

Exercise 2

定积分的数值计算

题目:

函数 $f(x)$ 在区间 $[a,b]$ 上的定积分可用区间内选取的 $n+1$ 个点 $x_i(i=0,1,\ldots,n)$ (称为积分节点) 上的函数值的加权和近似计算:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} w_i f(x_i)$$

其中 w_i 是函数值 $f(x_i)$ 的权重, 称为积分系数。常用的数值计算公式的区别体现在积分节点和积分系数上。

公式名称	积分节点和积分系数
复合梯形公式(Trapezoidal)	$x_i = a+ih \quad \text{for} \quad i=0,\ldots,n, \quad h=\frac{b-a}{n}$ $w_0 = w_n = \frac{h}{2}, \quad w_i = h \quad \text{for} \quad i=1,\ldots,n-1$
复合辛普森公式(Simpson), `n`必须为偶数，若输入奇数`n`，执行`n=n+1`	$x_i = a+ih \quad \text{for} \quad i=0,\ldots,n, \quad h=\frac{b-a}{n}$ $w_0 = w_n = \frac{h}{3}, \quad w_i = \frac{2h}{3} \quad \text{for} \quad i=2,4,\ldots,n-2$, $w_i = \frac{4h}{3} \quad \text{for} \quad i=1,3,\ldots,n-1$
复合高斯-勒让德公式(Gauss-Legendre), `n`必须为奇数，若输入偶数`n`，执行`n=n+1`	$x_i = a+\frac{i+1}{2}h-\frac{\sqrt{3}}{6}h \quad \text{for} \quad i=0,2,\ldots,n-1,$ $x_i = a+\frac{i}{2}h+\frac{\sqrt{3}}{6}h \quad \text{for} \quad i=1,3,\ldots,n$ $h=\frac{2(b-a)}{n+1}, \quad w_i=\frac{h}{2}, \quad \text{for} \quad i=0,1,\ldots,n$

源代码:

```
print("This is a python code for homework5-1: Integrator Class")
import numpy as np
import math

class Integrator(object):
    def __init__(self, a, b, n):
        self.a, self.b, self.n = a, b, n
        self.points, self.weights = self.compute_points()

    def compute_points(self):
        raise NotImplementedError('no rule in class %s' \
                                  % self.__class__.__name__)
    def integrate(self, f):
        sum = 0
        for i in range(self.n+1):
            sum = sum + self.weights[i]*f(self.points[i])
```

```
        return sum

def f(x): return x + 2
def F(x): return 0.5*x**2 + 2*x

class Trapezoidal(Integrator):
    def compute_points(self):
        p = list()
        w = list()
        for i in range(self.n+1):
            h = (self.b-self.a)/(self.n)
            p.append(self.a+i*h)
            if i == 0:
                w.append(h / 2)
            elif i == self.n:
                w.append(h / 2)
            else:
                w.append(h)
        return p, w

class Simpson(Integrator):
    def compute_points(self):
        p = list()
        w = list()
        if self.n%2 == 1:
            self.n = self.n+1
        for i in range(self.n+1):
            h = (self.b-self.a)/(self.n)
            p.append(self.a+i*h)
            if i == 0:
                w.append(h / 3)
            elif i == self.n:
                w.append(h / 3)
            elif i%2 == 0:
                w.append(2*h/3)
            else:
                w.append(4 * h / 3)
        return p, w

class GaussLegendre(Integrator):
    def compute_points(self):
        p = list()
        w = list()
        if self.n%2 == 0:
            self.n = self.n+1
        for i in range(self.n+1):
            h = 2*(self.b-self.a)/(self.n+1)
            if i%2 == 0:
                p.append(self.a+(i+1)*h/2-math.sqrt(3)*h/6)
```

```
            else:
                p.append(self.a+i*h/2+math.sqrt(3)*h/6)
            w.append(h/2)
        return p, w

# A linear function will be exactly integrated by all
# the methods, so such an f is the candidate for testing
# the implementations


def test_Integrate():
    """Check that linear functions are integrated exactly."""
    def f(x): return x + 2
    def F(x): return 0.5*x**2 + 2*x

    a = 2; b = 3; n = 4     # test data
    I_exact = F(b) - F(a)
    tol = 1E-6

    methods = [Trapezoidal, Simpson, GaussLegendre]
    for method in methods:
        integrator = method(a, b, n)

        I = integrator.integrate(f)
        if abs(I_exact - I)/I_exact > tol:
            print ('Error in %s' % method.__name__)

if __name__ == '__main__':
    test_Integrate()
```

运行结果：

```
/usr/local/bin/python3.9 "/Users/wangyijie/Library/Mobile
Documents/com~apple~CloudDocs/Study_in_USTC/杂事/python科学计算/hw5/hw5-2.py"

This is a python code for homework5-1: Integrator Class

Process finished with exit code 0
```

可见，测试函数没有报错，说明程序运行正常。