

Ayudantía N°8
Taller de Programación 2019
Diego Caro

1. Problema:

- **Objetivo:** implementar un tipo de dato para manejar números racionales (fracciones).
- **Conjunto de valores:** ... debe hacerlo usted...
- **Operaciones:**
 - suma, resta, multiplicación y división
 - devolver una representación en string

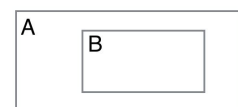
API:

class Rational

	Rational(n, m)	Crea un número racional de numerador n, y denominador m
Rational	plus(Rational b)	Suma de racional actual con b.
Rational	minus(Rational b)	Diferencia de racional actual con b.
Rational	times(Rational b)	Producto de racional actual con b.
Rational	divides(Rational b)	Cociente de racional actual con b.
Rational	simplify(Rational b)	Simplifica el racional b, si no es posible, retorna b.
str	__str__()	Representación en string del número racional

2.

- **Objetivo:** implementar un tipo de dato para manejar rectángulos en el plano. Extienda el código presentado a continuación.
- **Conjunto de valores:** el punto de la esquina inferior-izquierda y el punto superior-derecha del rectángulo.
- **Operaciones:** devolver el área, devolver True/False si dos rectángulos están solapados, devolver el área de intersección, devolver un nuevo rectángulo que sea el área de intersección.
- Proponga una **API** para las operaciones definidas. Rellene las operaciones y el retorno de cada operación en la siguiente tabla:



B está contenido en A



A y B están solapados. Área intersección.

class Rectangle

	Rectangle(...)	Crea un rectángulo..... (completar!)

str	<code>__str__()</code>	

```

1 class Point:
2     """A 2d point in (x, y)."""
3     def __init__(self, x, y):
4         self.x = x
5         self.y = y
6
7     def distX(self, p): return abs(self.x - p.x)
8     def distY(self, p): return abs(self.y - p.y)
9
10 class Rectangle:
11     """A rectangle with the lower-left corner in
12     p1 and upper-right corner in p2."""
13     def __init__(self, p1, p2):
14         self.p1 = p1
15         self.p2 = p2
16
17     def area(self):
18         return self.p1.distX(self.p2) * self.p1.distY(self.p2)
19
20 def main():
21     p1 = Point(-1, -1)
22     p2 = Point(1, 1)
23     r = Rectangle(p1, p2)
24     print('area de r:', r.area())
25
26 if __name__ == '__main__': main()

```

- Re-implementar el módulo de matrices de la Ayudantía 07 utilizando un nuevo tipo de datos (`class`). Diseñe una nueva API que tenga la misma funcionalidad del módulo matriz, e indique la tabla de operaciones que implementó en un archivo pdf (la vamos a leer para ejecutar su código).

Diseñe la API de tal forma que cada operación retorne una nueva matriz compuesta, usando una estrategia similar a la de la clase Rational.

Recuerde tomar en cuenta las pre y post condiciones que se deben cumplir para poder operar dos matrices. El código de la clase estar dentro de un módulo llamado algebra (es decir, un archivo llamado `algebra.py`).